
MODULO, DES MOYENS D'ENSEIGNEMENT DE L'INFORMATIQUE À VISÉE PARTICIPATIVE

Résumé. Cet article présente Modulo, des moyens d'enseignement participatifs pour l'enseignement de l'informatique au lycée dans le canton de Vaud en Suisse. Ces moyens sont basés sur une infrastructure informatique ouverte et versionable. Distribués sous une licence *Creative Commons*, ils peuvent être repris, modifiés et diffusés par les enseignant·e·s à l'image des logiciels libres. Ces ressources, présentées sous la forme d'un site web interactif, sont également imprimables pour un usage sous forme de manuel. Cet article décrit les choix techniques, pédagogiques et didactiques qui ont guidé la conception de ces ressources, afin de favoriser l'implication des élèves, et la libre adoption de ces moyens par les enseignantes et enseignants. L'objectif est d'initier une communauté de pratiques enseignantes autour de ces ressources permettant de les faire vivre et évoluer par les contributions des personnes actives sur le terrain.

Mots-clés. Ressources pédagogiques, communauté de pratique, inclusivité

Introduction

L'introduction d'une nouvelle discipline scolaire représente un changement marquant au sein de l'institution scolaire. Ce changement procède d'une recomposition qui se fait souvent au détriment d'autres disciplines qui voient leur dotation horaire ou leur domaine d'enseignement se réduire. Cela peut susciter certaines tensions au sein du corps enseignant soumis à des phénomènes de déqualification et de contraction du marché du travail. Des offres de formations continues permettant une requalification du personnel enseignant dans cette nouvelle discipline sont alors généralement proposées dans le but de pouvoir disposer de suffisamment d'enseignant·e·s susceptibles d'enseigner la nouvelle discipline (Harlé, 2010).

Les enseignant·e·s qui portent cette nouvelle discipline doivent alors la mettre en pratique dans les salles de classe, sans pouvoir s'appuyer sur leur propre vécu d'élève de cette discipline scolaire ni se référer à une tradition d'enseignement longuement établie, telle qu'elle existe dans les disciplines plus anciennes. Ceci dans un contexte où la légitimité de cette nouvelle discipline n'est pas forcément reconnue par les autres enseignant·e·s et les élèves, et où les enseignant·e·s requalifié·e·s peuvent également douter de la légitimité de leur propre enseignement.

C'est dans une telle situation que se trouve actuellement l'informatique, qui a été rendue obligatoire dans les lycées (également appelés gymnases) en Suisse en 2022. Afin de soutenir les enseignant·e·s de cette nouvelle discipline et favoriser sa constitution sur le terrain, une plateforme participative proposant des ressources d'enseignement a été créée par une équipe d'enseignant·e·s d'informatique en collaboration avec des institutions d'enseignement supérieur. Cette contribution, rédigée par un sous-ensemble de cette équipe, décrit le dispositif réalisé (accessible sur <https://modulo-info.ch>) et explicite les choix pédagogiques, didactiques et technologiques qui y ont mené.

1. Contexte institutionnel

En Suisse, l'Ordonnance sur la reconnaissance des certificats de maturité gymnasiale (ORM) fixe les conditions requises pour que les certificats de maturité (équivalents du baccalauréat en

France) délivrés par les cantons soient reconnus au niveau fédéral. En 2018, cette ordonnance a été modifiée avec l'ajout de l'informatique comme discipline obligatoire, au plus tard en 2022. En conséquence, le canton de Vaud (où se trouve Lausanne), comme les autres cantons suisses, a dû intégrer cette nouvelle discipline au plan d'études (ou curriculum) de l'École de Maturité (cursus suivi par environ un cinquième de la population qui permet l'accès à toutes les universités et écoles polytechniques du pays). Il a été décidé de remplacer l'ancien cours de bureautique par un cours d'informatique sur deux ans à raison d'une période (de 45 minutes) par semaine en classe entière (20-26 élèves) à laquelle s'ajoutent deux périodes de travaux pratiques en salle d'informatique toutes les deux semaines en demi-classe. Le plan d'études de ce cours, rédigé par des enseignant·e·s du canton, développe cinq domaines disciplinaires : représentation de l'information, programmation, algorithmique, architecture des ordinateurs et réseaux. Un domaine interdisciplinaire complète le tableau en abordant également les enjeux sociaux. Ce plan d'études n'a pas pour objectif de former des futur·e·s ingénieur·e·s en informatique, mais de former des citoyennes et citoyens à l'aise dans un monde de plus en plus numérisé. Comme pour d'autres disciplines, le plan d'études est volontairement formulé de manière assez ouverte, laissant ainsi aux enseignant·e·s un certain choix dans les sujets abordés. De même, une totale liberté est laissée aux enseignant·e·s quant aux supports de cours utilisés et beaucoup d'enseignant·e·s produisent leur propre matériel d'enseignement, le marché local des manuels étant trop restreint pour intéresser le secteur privé.

Toutefois, afin de faciliter la mise en place de cette discipline, les autorités politiques ont souhaité soutenir les futur·e·s enseignant·e·s de la discipline en mettant à disposition des moyens d'enseignement complets, pour lesquels un comité de rédaction a été constitué. Ces moyens sont principalement rédigés par une équipe enseignante d'informatique en collaboration avec des institutions d'enseignement supérieur (École polytechnique fédérale de Lausanne, Université de Lausanne et Haute école pédagogique du canton de Vaud). Ceci permet d'avoir une partie de l'équipe de rédaction proche du terrain et partageant les préoccupations des enseignant·e·s à qui ces moyens sont destinés.

2 Cahier des charges

Les autorités politiques ont exprimé deux souhaits. Le premier souhait était de rendre cette nouvelle discipline accessible à tous les élèves, indépendamment de leur orientation, sans qu'elle ne génère un important échec scolaire. Comme cette nouvelle discipline s'adresse à des élèves aux profils extrêmement variés, il a fallu être particulièrement attentifs à ne pas pénaliser davantage les élèves d'orientation non scientifique.

Le second souhait était d'intégrer les contenus informatiques dans un contexte social plus global. Si l'enseignement de la science informatique a pour objectif de transmettre des connaissances de base dans des domaines techniques tels que l'algorithmique ou la programmation, la numérisation croissante de la société nécessite également la compréhension des enjeux sociaux, politiques et économiques que ces transformations soulèvent. Une certaine hauteur de vue est nécessaire pour saisir la façon dont le numérique se déploie dans la société. C'est pourquoi les ressources proposées visent une approche nuancée du numérique, qui ne relève ni d'un techno-optimisme naïf, ni d'une techno-critique qui se limiterait à souligner les risques des technologies. À ce titre, les outils des sciences humaines s'avèrent particulièrement utiles pour saisir la complexité des enjeux et développer une pensée critique. Cette dernière recouvre une capacité à construire, situer et contextualiser une problématique, afin de comparer et évaluer les différents choix potentiels qui déterminent à chaque instant notre quotidien, aussi bien individuellement

que collectivement. C'est pourquoi des enseignants de sociologie exerçant tant au niveau du post-obligatoire qu'au niveau universitaire ont été impliqués dans le projet.

3. Nature des ressources

Si le mandat donné à l'équipe de rédaction était de rédiger des moyens d'enseignement pour soutenir les enseignant·e·s d'informatique, la forme que devaient prendre ces moyens n'était pas spécifiée. Il est vite apparu risqué d'opter pour un manuel scolaire traditionnel publié par un éditeur et vendu aux élèves. En effet, les enseignant·e·s de gymnase choisissant librement leurs supports de cours (ou les produisant eux-mêmes), pourraient être réticent·e·s à faire acheter à leurs élèves un manuel avant d'avoir eu l'occasion de l'essayer et dont il n'est pas sûr qu'il leur convienne. De plus, le recours à un éditeur privé pourrait, pour des questions de droits d'auteur, entraver une libre appropriation et diffusion du manuel par les enseignant·e·s, augmenter son prix de vente envers les élèves, limitant ainsi son utilisation. Enfin, l'informatique est l'objet dans le canton de Vaud d'une réforme également dans les degrés inférieurs, et il est possible que le plan d'étude évolue pour s'adapter à une prochaine introduction plus précoce de l'informatique à l'école.

Des moyens d'enseignement en ligne, par exemple sous forme de site web, permettent une plus grande adaptabilité et un cycle de ré-édition bien plus rapide que sous forme imprimée. Il nous a semblé utile de pouvoir bénéficier d'une mise à jour tous les six mois afin de pouvoir rapidement corriger et adapter ces moyens à la lumière des expériences faites lors de son utilisation dans des conditions réelles. Par ailleurs, des ressources en lignes peuvent être interactives et donc apporter une plus-value pédagogique, par exemple l'exécution de code, des questionnaires auto-corrigés, des animations multimédia ou des liens sur des ressources externes (Bruillard & Baron, 1998). C'est pourquoi nous avons opté pour un format web, illustré sur la Fig 1. Toutefois, il nous a semblé également utile que ces ressources puissent être disponibles sur support papier, d'une part car tous les élèves n'ont pas forcément accès à internet à domicile et en déplacement, d'autre part car les cours ont partiellement lieu dans des salles de classe dépourvues d'ordinateur. La génération automatique d'une version imprimable des ressources est donc également possible. Celle-ci n'est toutefois pas une simple impression des pages HTML, le support papier imposant d'autres contraintes de mise en page et d'interactivité.

De même, afin de permettre aux enseignant·e·s d'adapter ces moyens à leur convenance, il a été décidé d'en publier les sources ouvertement sous une licence *Creative Commons*. Ainsi, les enseignant·e·s ont le droit de modifier ces moyens et sont libres de les partager ensuite. Cela permet d'élargir la base d'enseignant·e·s susceptibles d'utiliser les ressources aux personnes qui ne souhaitent pas les utiliser exactement telles que proposées, tout en favorisant les échanges autour de leur utilisation.

Il a été décidé de distinguer deux parties, comme illustré sur la Figure 1. :

- **Une partie “Apprendre”**, destinée aux élèves et présentant les contenus d'apprentissage ainsi que des exercices (Figure 2).
- Une partie “Enseigner”, destinée aux enseignant·e·s et proposant des pistes pédagogiques, des activités à faire en classe, des séquences d'enseignement, des exemples de planifications annuelles et du contenu sur les enjeux sociaux de l'informatique (Figure 3).

L'enseignement des enjeux sociaux de l'informatique est certainement l'aspect le plus susceptible de mettre les enseignant·e·s d'informatique en difficulté. Pour faire face à leurs éventuelles craintes et réticences, ces ressources visent à leur fournir à la fois des clés de lecture des principaux enjeux sociaux du numérique et des pistes pédagogiques pour les mettre en pratique. Deux formats de ressources sont proposés :

- **Des grands dossiers thématiques**, qui correspondent aux sept sujets “enjeux de société” figurant dans le plan d'études vaudois de la discipline “Informatique”.
- **Des “fiches complémentaires”**, qui permettent d'enrichir les dossiers thématiques en éclairant une notion, un événement ou une controverse.

Ces deux formats ont été pensés avec l'objectif d'être complémentaires et modulables. Les thématiques ne sont pas cloisonnées et peuvent être aisément combinées, au vu de la proximité de certains enjeux. Par exemple, la question des modèles économiques du numérique amène nécessairement à évoquer la problématique de la surveillance (Abiteboul & Doweck, 2017).

4 Approches pédagogiques et didactiques

Au-delà de sa potentielle « efficacité », une approche pédagogique ou didactique donne une coloration particulière à la discipline enseignée. Dans le même sens que la volonté politique présidant à la rédaction de ces ressources, il nous a paru essentiel de choisir des approches pédagogiques et didactiques qui s'adressent au plus grand nombre possible d'élèves, afin que toutes et tous puissent s'engager avec confiance dans la découverte de l'informatique. Ceci est d'autant plus important que l'informatique est déjà bien chargée de représentations chez les adolescents, notamment autour de la culture geek et des jeux vidéos (Varma, 2007). Afin de favoriser l'engagement des élèves et rendre cette discipline aussi inclusive que possible, les approches pédagogiques et didactiques suivantes ont été adoptées.

1. Une prise de distance par rapport aux mathématiques scolaires. En effet, les mathématiques participant de manière significative à l'échec scolaire, elles sont souvent associées à des représentations négatives chez les élèves, représentations qui influencent leur capacité à s'engager avec succès dans des tâches cognitives associées (Huguet et al., 2001). C'est pourquoi nous avons tenté une « dé-mathématisation » de l'informatique. Par ceci, nous entendons que les formalismes sont autant que possible évités, de même que les exemples d'application tirés des mathématiques. De plus, beaucoup d'exemples classiques d'algorithmes et de programmation issus du domaine mathématique, tels que la recherche du plus grand dénominateur commun, le calcul de la racine carrée, ont volontairement été évités afin de ne pas induire un rejet de la part de certains élèves mal assurés dans cette discipline. À la place, les ressources essaient de s'appuyer sur les aspects concrets de l'informatique et l'expérience quotidienne des élèves afin de développer des intuitions correctes et des représentations adéquates. Par exemple, la partie programmation est basée sur la pédagogie constructionniste de Seymour Papert (Papert, 1980), avec l'utilisation de la tortue logo (par le module turtle de Python) pour faire des dessins. Plutôt que de nommer les chapitres directement avec les objets didactiques comme c'est l'usage, les concepts introduits sont plutôt référencés par l'action qu'elles permettent de réaliser. Par exemple, à la place d'avoir des chapitres “instruction”, “variables”, “fonctions”, ces notions sont abordées respectivement dans les chapitres “dessiner”, “mémoriser” et “définir”. Au lieu de partir avec des concepts techniques et parfois obscurs pour les élèves du langage de programmation (par exemple « instruction conditionnelle » ou « fonction de rappel »), nous nous sommes inspirés des activités de tous les jours (décider,

Modulo est un catalogue de ressources destiné à l'enseignement et à l'apprentissage de l'informatique au gymnase.

Figure 1: Page d'accueil du site Modulo.

déclencher), que nous explorons à travers un programme informatique. Ceci n'empêche pas que chaque chapitre correspond à un concept informatique bien défini qui sera réinvesti par la suite. Les exercices proposés en programmation permettent aux élèves d'exprimer leur créativité non seulement dans la résolution de problèmes informatiques mais également dans la réalisation d'œuvres personnelles sous la forme de dessins complexes.

De même, la partie concernant la représentation de l'information traite de la représentation du son sans s'appuyer sur les fonctions trigonométriques représentant un son pur, mais en abordant la notion peut-être plus intuitive d'échantillonnage (manipulée dès la petite enfance avec les activités classiques de points à relier pour réaliser un dessin) permettant le passage du continu au discret, une distinction fondamentale de l'informatique. Pour la notion de redondance, l'accent n'est pas mis sur les opérations spécifiques, mais sur le concept général qui peut être décliné de manière très concrète. Cette « dé-mathématisation » reste bien entendu limitée à la présentation, l'objectif étant uniquement qu'un langage mathématique ne vienne troubler l'engagement des élèves, même si les raisonnements et les habiletés cognitives exercées restent de type mathématique. Nous sommes conscients de l'importance des mathématiques et de leur rôle central dans l'informatique, mais nous pensons (et observons dans une certaine mesure) que certains élèves, à ce stade de leur parcours, rentrent plus facilement dans des activités mathématiques par la porte arrière, en s'y adonnant sans le savoir, jusqu'à le découvrir par eux-mêmes.

2. Une pédagogie orientée sur la découverte et l'intuition. Par exemple, les parties Programmation et Architecture des ordinateurs proposent les deux des activités liées à la création et l'exécution de programmes et de systèmes logiques, respectivement. Mais dans un cas comme dans l'autre, la compréhension intuitive via une expérimentation extensive de programmes et systèmes fonctionnant déjà est favorisée: l'écriture de programmes ou la composition de petits circuits logiques elles-mêmes interviennent dans un second temps. En ce sens, nous suivons une approche de type *Use-Modify-Create* (Lee et al., 2011).

3. Un recours à des activités de collaboration. Comme d'autres sciences, le développement de l'informatique est une œuvre collective qui a nécessité d'importants niveaux de collaboration. L'informatique a peut-être poussé cette notion plus loin que d'autres disciplines avec le développement des logiciels libres et la création du Web, conçu originalement comme un outil de collaboration scientifique (Cardon, 2019). L'architecture d'Internet, conçu comme moyen de communication décentralisé à la gouvernance collaborative et non hiérarchique est aussi

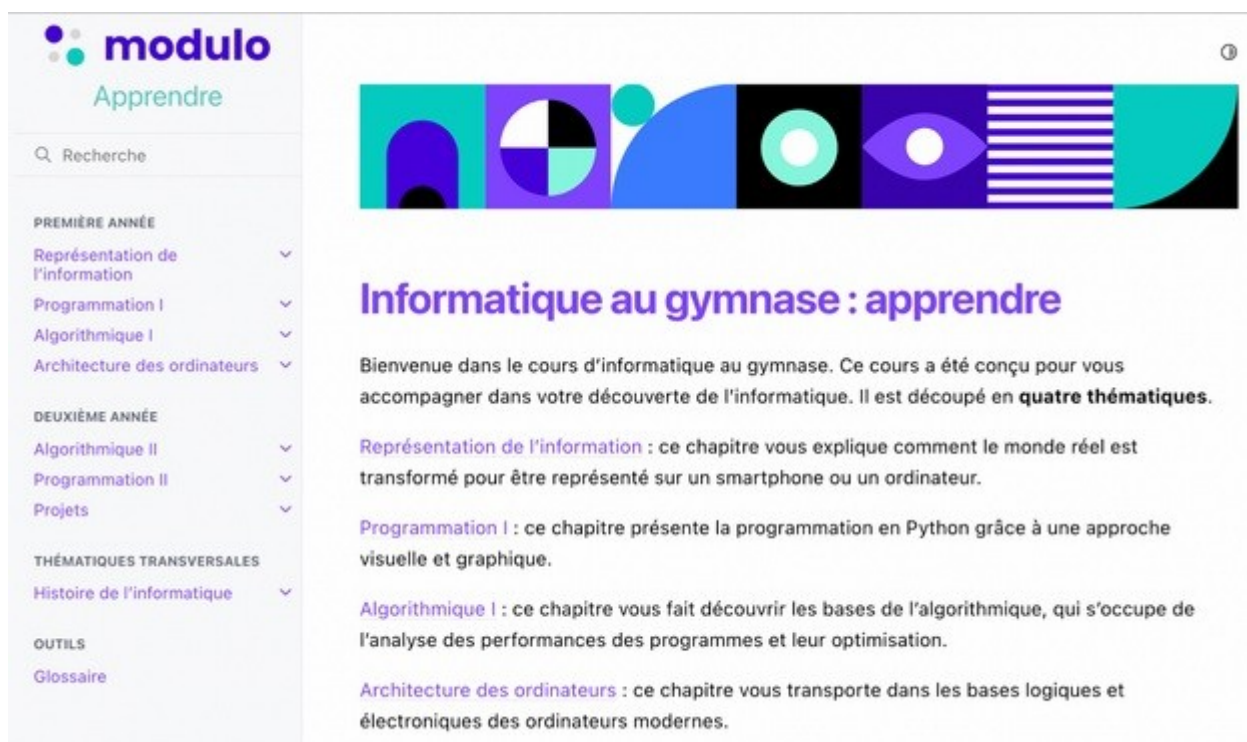


Figure 2: page d'accueil de la partie "Apprendre" destinée aux élèves.

révélatrice à cet égard. La collaboration influençant de plus positivement la dynamique motivationnelle des élèves (Viau, 2004), les ressources présentent également des activités collaboratives ou collectives, telles que l'élaboration d'un protocole de communication, la simulation d'enchères publicitaires sur une plateforme ou la conception collaborative d'un algorithme.

4. Une ouverture sur un horizon culturel diversifié. En particulier, nous avons pris soin d'éviter une présentation trop genrée de l'informatique dans laquelle les éléments de la culture geek ou du monde des jeux vidéos seraient prépondérants. Nous avons au contraire inclus des références à l'art contemporain et à la culture populaire. Par exemple en programmation, l'univers visuel des exemples puise dans les objets de tous les jours (maison, chaise, table, lit) mais aussi dans drapeaux des pays, les jeux vidéo classiques (Tetris, Minecraft, Pokemon) les couleurs marquées (DeepPink, Lime, Aqua) ou encore des émojis et leurs émotions. De plus, les ressources présentent des personnages importants de l'histoire de l'informatique, sans en invisibiliser les femmes. L'objectif est de lutter contre une division sexuée des savoirs (Mosconi, 2016) dans laquelle les garçons occupent certains domaines du savoir plus socialement valorisés et que les filles se sentent moins légitimes à investir.

Des pratiques à la théorie

Un consensus ayant été atteint sur ces choix pédagogiques et didactiques, ces derniers ont été appliqués de manière différentes par les rédactrices et rédacteurs des différents chapitres. La documentation de leurs intentions didactiques et pédagogiques s'est révélée nécessaire pour une appropriation et une utilisation pertinentes des ressources en classe par les enseignant·e·s. Ce faisant, les rédactrices et rédacteurs ont été incités à documenter leur bonnes pratiques professionnelles, issues de leur expérience, sans que celles-ci n'aient été nécessairement



Figure 3: Page d'accueil de la partie "Enseigner" destinée aux enseignant·e·s.

formalisées dans le cadres de la recherche académique en didactique. Pour le chapitre d’algorithmique, cela s’est traduit par le cadre théorique des “**Cinq Co**” qui nous a guidé pour la conception des contenus théoriques et pratiques. Ce cadre consiste en cinq composantes (voir Figure 4) qui commencent par des aspects généraux pour se rapprocher de plus en plus vers l’élève.

1. **Cohérence**, ou raconter une histoire cohérente. Le choix d’aborder l’algorithmique par les *algorithmes de tri* a été motivé par l’envie de raconter une histoire unique et cohérente, une sorte de *fil rouge* multi-usages, qui couvrirait tous les objectifs pédagogiques. En effet, même si les algorithmes de tri peuvent paraître fastidieux au premier abord, ils permettent d’aborder non seulement la *pluralité des algorithmes* (e.g. « tri par sélection », « tri par fusion »), mais aussi les *stratégies algorithmiques plus complexes* (e.g. « diviser pour régner ») qui figurent au plan d’études.
2. **Concepts et principes**, ou éviter les formules mathématiques. Nous avons présenté ci-dessus les raisons derrière notre souhait de dé-mathématiser l’informatique. Démathématiser un chapitre comme l’algorithmique représentait un défi de par la nature des objets didactiques. Pour y arriver, nous nous sommes focalisés sur les concepts et sur les principes, en les formalisant le moins possible. Par exemple, nous avons préféré nous appuyer sur la représentation intuitive de ce qu’est un tri dans la vie courante, plutôt que de commencer par une définition mathématique du tri. A la fin de la séquence pédagogique, les élèves sont capables d’appliquer, coder et comparer objectivement plusieurs algorithmes de tris différents.
3. **Contextualisation**, ou faire des références à la vie courante, aux autres disciplines et à des petits casse-têtes. Quand cela était possible, nous avons utilisé des exemples qui

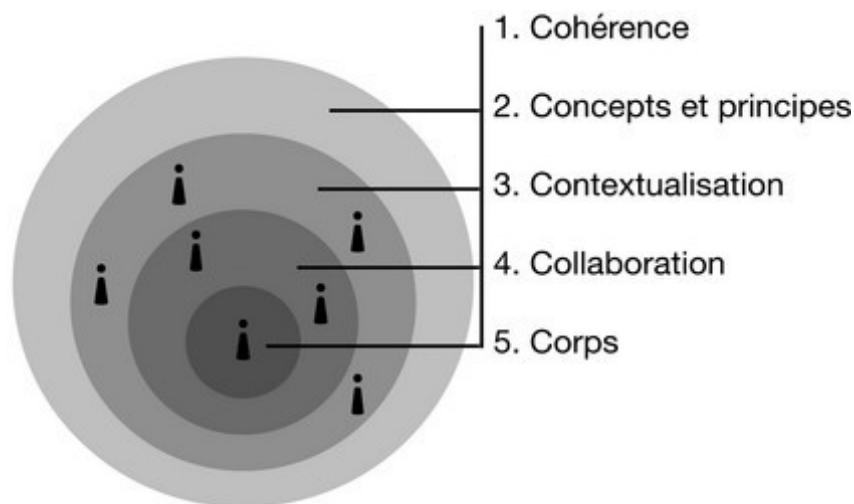


Figure 4: Le cadre de "cinq Co".

s'inspirent de la vie courante, comme chercher son chemin ou organiser un voyage ou une fête d'anniversaire. Cela éveille l'intérêt des élèves, qui s'impliquent davantage lorsqu'ils voient l'utilité d'une solution de problème (Viau, 2004). A titre d'exemple, après avoir abordé les expressions booléens avec un exemple de météo, un élève s'est exclamé pour dire qu'il venait enfin de « comprendre le cours de maths de l'année dernière sur la logique ». En tant qu'outil de communication, d'investigation scientifique ou d'expression artistique, l'informatique touche à toutes les disciplines et se prête bien à des activités interdisciplinaires, permettant de « faire découvrir la complexité de la réalité en développant l'esprit critique par la confrontation de points de vue différents ou par la multiplicité des approches » et à « favoriser la réactivation de notions et de méthodes déjà vues dans les cours, ainsi que leur enracinement dans des sujets particuliers » (CDIP, 2020). C'est pourquoi nous proposons des activités en lien avec d'autres disciplines, par exemple en lien avec les arts visuels (e.g. algorithme dans la tête de Mondrian), la biologie (e.g. modélisation d'une épidémie) ou encore l'économie (e.g. les enchères du Web). Finalement, nous entrons régulièrement dans la matière par des problèmes à résoudre sous la forme d'encarts intitulés « matière à réfléchir » (généralement en début de chapitre) et par des petits casse-têtes, dans l'idée de stimuler la réflexion des élèves et influencer positivement sur leur motivation (Viau, 2004).

4. **Collaboration**, ou travailler en équipe. En plus de faire partie intégrante du développement des objets informatiques (voir ci-dessus), le travail en équipe sur des grands projets est la norme en informatique. L'évaluation par projets fait partie intégrante du parcours d'éducation supérieure d'un·e informaticien·ne. Les compétences de planification, d'anticipation et d'organisation nécessaires pour mener à bien un projet nous semblent également importantes pour le travail de maturité que les élèves réalisent en dernière année. Pour toutes ces raisons, nous avons souhaité que les ressources proposées reflètent cette réalité du terrain et intègrent l'esprit d'équipe, en suggérant à l'élève de confronter sa solution avec celle de ses camarades ou de résoudre des problèmes en groupe.
5. **Corps**, ou manipuler des objets et bouger. En contextualisant, nous avons souhaité faire

réfléchir les élèves avec leur tête, mais pas seulement. En effet, conformément aux théories énaactives de la cognition (Pfeifer & Bongard, 2006; Varela et al., 1991), les aspects cognitifs et sensori-moteurs sont étroitement couplés. L'engagement du système moteur dans les phases d'apprentissage, par exemple par la manipulation manuelle d'objets (e.g. dispositif physique d'algorithmes de tri) ou le mouvement du corps dans l'espace (e.g. activité Dijkstra et le plus court chemin) permet d'associer plus de modalités sensorielles et motrices aux objets d'apprentissage et favorise leur intégration, un phénomène bien étudié (et débattu) (Pouw et al., 2014), mais peu mis en pratique au niveau du lycée.

5. Choix technologiques

Le choix s'étant vite porté sur le développement d'un site web, la question s'est posée de l'outil à utiliser pour la génération de ce site. Il nous a paru nécessaire d'avoir un support facilement éditable afin que sa modification soit relativement aisée et rapide pour des enseignant·e·s d'informatique. De plus, afin de garantir une certaine indépendance et résilience des moyens vis-à-vis des logiciels utilisés, il paraissait important de se baser uniquement sur des outils open-source sous licence libre. Ceci permet de se protéger de décisions de la part d'un éditeur de logiciel qui mettrait en péril le maintien des moyens que ce soit pour des questions financières, de compatibilité ou de fonctionnalité. Enfin, afin d'exploiter les possibilités d'interactivité offertes par le Web, il fallait un outil suffisamment flexible, permettant notamment l'exécution de code en ligne et l'implémentation de fonctionnalités spécifiques. Par ailleurs, il nous semblait également utile d'offrir la possibilité de générer une version imprimable du site, les élèves n'ayant pas forcément un accès à internet à domicile.

Quelques essais ont été effectués avec Jupyter-book, une émanation de Jupyter notebook largement utilisée dans le domaine académique pour créer des documents dans lequel on peut exécuter du code. Dans la pratique, toutefois, ceci nécessite le recours à des serveurs Jupyter qui sont relativement lents (pour les versions gratuites) ou qu'il faut installer localement à l'école. Notre choix s'est finalement porté sur Sphinx (Brandl, s. d.), le moteur de documentation sur lequel est basé Jupyter-book qui permet une plus grande flexibilité. Sphinx, développé initialement pour générer de la documentation de code, génère un site web statique (côté serveur) à partir de fichiers sources rédigés en syntaxe Markdown, similaire à la syntaxe de Wikipedia. Sphinx permet également la génération d'un document pdf (généré par LaTeX) qui peut être utilisé comme polycopié et distribué aux élèves. Ainsi, à partir des sources en markdown, il est possible de générer automatiquement à la fois un site web statique au format HTML et CSS facilement déployable sur un serveur et un document pdf imprimable.

Il est de plus possible de programmer des extensions à Sphinx, par exemple pour définir de nouvelles balises en Markdown, ou proposer des contenus spécifiques à la version web ou la version pdf. Nous avons donc développé plusieurs extensions, permettant en particulier l'exécution de programmes Python directement dans le navigateur (du côté client, grâce à la bibliothèque Skulpt [<http://skulpt.org/>], une implémentation de Python en JavaScript), des questions à choix multiples auto-corrigées, l'insertion de vidéos, la constitution d'un glossaire, et la simulation interactive et construction de circuits logiques [Pellet 2022].

Dans le but de faciliter les collaborations, les sources de la plateforme et de ses contenus sont hébergés et librement accessibles sur GitHub, une populaire plateforme d'hébergement de code.

La plateforme GitHub repose sur git, un logiciel de gestion de version largement adopté dans le monde du développement logiciel. Toute personne qui le souhaite peut obtenir les sources du projet afin de monter sa propre version de la plateforme, et proposer des suggestions ou des modifications. Ceci permet aux enseignant·e·s d'adapter les ressources à leur goût, d'en générer une version spécifique à leur école (ou canton) et de proposer des ajouts ou des modifications pour faire évoluer la version officielle des ressources. La version officielle est mise à jour tous les ans (sauf erreurs manifestes ou coquilles), mais la version en développement appelée à devenir la prochaine version officielle est également disponible, permettant aux enseignant·e·s d'anticiper les changements.

6. Communauté

Au delà des moyens d'enseignements eux-mêmes, le fait de proposer des ressources ouvertes, que chaque enseignant·e peut modifier, publier et s'approprié à sa manière vise également à promouvoir les échanges autour de l'enseignement de l'informatique et contribuer à l'émergence et à la vivacité d'une communauté de pratiques (Wenger, 1999). En plus de développer les compétences (Gueudet & Trouche, 2008) et d'asseoir l'identité professionnelle des enseignant·e·s d'informatique, une telle communauté sera utile pour forger l'informatique comme discipline scolaire et homogénéiser en partie les pratiques enseignantes. L'équipe de rédaction, qui s'est régulièrement réunie pendant deux années, peut constituer un noyau parmi d'autres de cette communauté de pratiques, tout comme des enseignant·e·s ayant accepté de tester les ressources lors d'une phase pilote l'année dernière. De plus, les formations et les rencontres organisées autour de ces ressources permettent d'inclure davantage d'enseignant·e·s dans ces échanges, qui peuvent également se poursuivre en ligne sur un forum de discussion dédié. Enfin, un suivi scientifique sous forme d'enquêtes et entretiens auprès des enseignant·e·s et des élèves, documentant comment ces ressources sont prises en main et perçues par les élèves et les enseignant·e·s est également mis en place et permettra d'alimenter la réflexion autour de l'enseignement de l'informatique et de l'appropriation de ressources d'enseignement libres.

Il est pour l'instant prématuré pour une évaluation précise des ressources, mais les premiers retours que nous avons reçus sont globalement positifs.

Il a été rapporté que des élèves rentrent facilement dans l'approche visuelle de la programmation et restent souvent en classe pendant la pause, pour continuer à programmer, et à essayer de nouvelles choses. Plusieurs enseignant·e·s ont indiqué que même sans les utiliser tels quels, ces moyens d'enseignement leur sont utiles pour avoir une idée plus précise des contenus figurant au plan d'études, des niveaux d'exigences proposés et des compétences à exercer.

Lors d'un sondage auprès de tous les étudiant·e·s débutant cette année leur formation d'enseignant·e en informatique en Suisse Romande, toutes les personnes formées dans le canton de Vaud indiquent Modulo comme la première ressource d'enseignement dont ils ont connaissance ou qu'ils utilisent. Certains étudiants d'autres cantons francophones citent également cette ressource pour leur enseignement. Cela indique que les ressources Modulo sont systématiquement proposées aux enseignant·e·s d'informatique nouvellement arrivé·e·s dans les écoles vaudoises par leurs collègues ou formateurs de terrain. Ces ressources se disséminent également dans d'autres cantons francophones ayant un plan d'étude légèrement différent.

Les ressources Modulo permettent plusieurs niveaux de participation, que ce soit en les utilisant,

en partageant ses expériences, en proposant des corrections et des nouveaux contenus ou en relisant et validant les corrections proposées. Une gouvernance de ces ressources est nécessaire afin de les maintenir vivantes à l'issue de la phase de rédaction.

Une particularité de Modulo, par rapport à d'autres catalogues de ressources d'enseignement de l'informatique est que le projet a un soutien

Modulo est le fruit d'une collaboration entre des enseignant·e·s qui ont été partiellement déchargé de leur activité enseignante pour la création des ressources en collaboration du personnel d'institutions académiques. En ce sens, ce projet se distingue d'autres ressources partagées qui sont souvent soit l'initiative bénévole de quelques enseignant·e·s ou d'associations (potentiellement financées par le secteur privé) soit des initiatives du monde académique. Cette collaboration permet de faire se rencontrer une expertise théorique du monde académique et une expertise pratique du terrain dans une gouvernance horizontale où les décisions sont le plus souvent prises par consensus.

8. Conclusion

Selon (Choppin, 2005), le manuel scolaire remplit en général quatre fonctions: la fonction curriculaire détermine les contenus d'enseignement; la fonction instrumentale met en œuvre une certaine pédagogie; la fonction idéologique et culturelle promeut certaines valeurs, propose un certain regard et met en avant certains éléments de culture; la fonction documentaire, plus spécifique, expose certains documents sur lesquels les élèves exercent leur esprit critique et d'analyse. Les ressources Modulo n'échappent pas à la règle, bien que la fonction documentaire soit encore peu présente. Pour l'instauration d'une nouvelle discipline, les fonctions curriculaire, instrumentale, et idéologique-culturelle du manuel sont susceptibles de façonner de manière durable le visage de cette discipline scolaire.

En effet, le développement d'une discipline scolaire ne se réduit pas à ses contenus (Astolfi, 2008), mais inclut également des rapports au savoir, des modes d'accès au savoir, une culture et des valeurs. Dès ses origines, l'informatique a développé une forte culture de l'expérimentation, de la collaboration, et du partage, mais aussi de l'assujettissement (de la machine, mais pas seulement)(Collet, 2019). Ces ressources se posent en héritières critiques de cette culture et espèrent contribuer à forger une discipline scolaire que chacun et chacune se sent à l'aise d'investir et permettant au plus grand nombre de mieux comprendre le monde qui l'entoure.

Cette vision ne peut toutefois pas être imposée hiérarchiquement, même si elle peut être institutionnellement soutenue. Il est trop tôt pour savoir si ces ressources et cette vision trouveront un écho ou convaincront les enseignant·e·s d'informatique. La discipline scolaire informatique est encore en cours de constitution dans le canton de Vaud et son devenir reste ouvert. Mais la cinquième fonction des moyens d'enseignement Modulo, que l'on pourrait qualifier de fonction communautaire, vise à développer, par la participation, une communauté active qui orientera le futur de cette discipline.

Remerciements

Les auteurs et autrices remercient les membres du projet modulo qui ont rédigé et relu les ressources, développé la plateforme modulo, et coordonné le projet: <anonymisation>

Bibliographie

- Abiteboul, S., & Dowek, G. (2017). *Le temps des algorithmes*. Le pommier.
- Astolfi, J. P. (2008). *La saveur des savoirs: disciplines et plaisir d'apprendre*. ESF.
- Brandl, G. (s. d.). *Sphinx documentation*. <https://www.sphinx-doc.org/en/master/>
- Bruillard, E., & Baron, G.-L. (1998). Vers des manuels scolaires électroniques? Résultats d'une étude en mathématiques en classe de sixième. *Sciences et techniques éducatives*, 5(4), 343–370.
- Cardon, D. (2019). *Culture numérique*. Presses de Sciences Po.
- CDIP. (2020). *Evolution de la maturité Fédérale. Projet Actualisation du Plan d'études cadre*. CDIP. <https://matu2023.ch/fr/groupe-de-projet-et-de-travail/plan-d-etudes-cadre>
- Choppin, A. (2005). L'édition scolaire française et ses contraintes: une perspective historique. *Manuels scolaires, regards croisés*, 39–53.
- Collet, I. (2019). *Les oubliées du numérique*. Le Passeur.
- Gueudet, G., & Trouche, L. (2008). Du travail documentaire des enseignants: genèses, collectifs, communautés. Le cas des mathématiques. *Education et didactique*, 2-3, 7–33.
- Harlé, I. (2010). *La fabrique des savoirs scolaires*. La Dispute.
- Huguet, P., Brunot, S., & Monteil, J. M. (2001). Geometry versus drawing: Changing the meaning of the task as a means to change performance. *Social Psychology of Education*, 4(3), 219–234.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37.
- Mosconi, N. (2016). Excellence et égalité. *Nouvelles Questions Feministes*, 35(1),

118-130.

- Papert, S. (1980). *Mindstorms: Computers, children, and powerful ideas*. NY: *Basic Books*, 255.
- Pfeifer, R., & Bongard, J. (2006). *How the body shapes the way we think: a new view of intelligence*. MIT press.
- Pouw, W. T. J. L., van Gog, T., & Paas, F. (2014). An Embedded and Embodied Cognition Review of Instructional Manipulatives. *Educational Psychology Review*, 26(1), 51-72. <https://doi.org/10.1007/s10648-014-9255-5>
- Varela, F., Thomson, E., & Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. MIT Press.
- Varma, R. (2007). Women in Computing: The Role of Geek Culture. *Science as Culture*, 16(4), 359-376. <https://doi.org/10.1080/09505430701706707>
- Viau, R. (2004). La motivation: condition au plaisir d'apprendre et d'enseigner en contexte scolaire. *3e congrès des chercheurs en Éducation, Brussels, Belgium*.
- Wenger, E. (1999). *Communities of practice: Learning, meaning, and identity*. Cambridge university press.