



Résumé de l'article de Pierre Tchounikine du 20 janvier 2016 « *Initier les élèves à la pensée informatique et à la programmation avec Scratch* » <mailto:Pierre.Tchounikine@imag.fr>

Introduction.



Ce document a été conçu par Pierre Tchounikine, professeur d'informatique à l'université, afin d'éclairer les enseignants et formateurs sur l'enseignement de la « pensée informatique » d'une part et d'autre part sur « l'utilisation de l'environnement de programmation Scratch. Il s'agira ici de mettre au cœur de notre réflexion et nos pratiques didactiques, l'algorithmique et la programmation que l'on pourra soit inscrire dans une approche dite traditionnelle, soit orientée « informatique créative ». Le support technique utilisé sera l'environnement de programmation Scratch qui est pour le moment le mieux adapté au niveau de l'école primaire et plus particulièrement celui du cycle 3 (CM1/CM2/6^{ème}).

Notre étude s'articulera autour des questions suivantes : Qu'est-ce que la pensée informatique ? Comment et pourquoi enseigner l'informatique à l'école primaire ? Quels objectifs pédagogiques considérer ? Comment intégrer le langage de programmation Scratch dans cet enseignement ? Quelle posture l'enseignant doit-il adopter ? Quelles sont les difficultés que l'on risque de rencontrer ?

1- Qu'est-ce que la pensée informatique ?

C'est un ensemble d'attitudes et de connaissances universellement applicables, que nous gagnerions toutes et tous à apprendre et à maîtriser, et que nous devrions transmettre à nos enfants. Dit autrement, c'est une manière d'organiser sa pensée, permettant de mettre en œuvre des notions et des procédés permettant de la rendre explicite par le biais de l'informatique. Elle se réalise essentiellement par la mise au point d'un algorithme.

Un algorithme est un enchaînement mécanique d'actions, dans un certain ordre, qui chacune a un effet, et dont l'exécution complète permet de résoudre un problème ou de faire quelque chose

Un programme est un algorithme traduit dans un langage de programmation.

Un ordinateur est une machine qui (entre autres) est capable d'exécuter un programme, i.e., de réaliser les actions du programme.

Le langage Scratch permet cette approche :

- En appréhendant un problème et sa solution à différents niveaux (abstraction)
- En réfléchissant aux tâches à accomplir sous forme de série d'étapes (algorithmes)
- En comprenant qu'un problème complexe peut être décomposé en problèmes simples (décomposition)
- En s'appuyant sur les compétences déjà acquises par l'élèves (problèmes déjà résolus auparavant)
- En intégrant que la solution d'un problème peut permettre la résolution de tout un éventail de problèmes similaires (généralisation)

Les écueils à éviter :

- Ne pas confondre l'utilisation d'un ordinateur, d'un logiciel (type compétences du B2i) avec l'enseignement de l'informatique.
- Ne pas confondre l'informatique pour enseigner (exemple d'un logiciel de géométrie dynamique) avec l'enseignement de notions relevant de l'informatique)
- Le développement de la pensée informatique peut se faire sans ordinateur, notamment avec des activités dites « débranchées » articulées autour d'étapes intermédiaires dans la résolution d'un problème plus complexe.

2- Pourquoi enseigner l'informatique à l'école primaire ?

Les arguments en faveur :

- Favoriser la pensée scientifique en développant une sensibilisation précoce aux grands concepts de la science et des techniques informatiques (langage, traitement de l'information, programmation)
- L'initiation à la programmation est une clé d'entrée dans la réalisation d'activités créatrices (une même tâche pourra être accomplie de différentes manières, avec des habillages différents)
- Le socle commun de connaissances, de compétences et de culture y fait référence : « [L'élève] sait que des langages informatiques sont utilisés pour programmer des outils numériques et réaliser des traitements automatiques de données. Il connaît les principes de base de l'algorithmique et de la conception des programmes informatiques. Il les met en œuvre pour créer des applications simples. ». Le domaine 1 s'intitule « Les langages pour penser et communiquer... Comprendre, s'exprimer en utilisant les langages mathématiques, scientifiques et informatiques ».
- Les programmes officiels du cycle 3 y font référence :

Plusieurs références au « langage de programmation » en lien avec les mathématiques sont également formulées (construction de figures et déplacements). En sciences, il est précisé que « Les élèves découvrent l'algorithme en utilisant des logiciels d'application visuelles et ludiques. Ils exploitent les moyens informatiques en pratiquant le travail collaboratif ». Les nouveaux programmes de l'école primaire évoquent donc une initiation à l'informatique, sans plus de précision cependant, prioritairement tournée vers les mathématiques.

Ainsi, les compétences développées dans l'enseignement de l'informatique peuvent être utiles aux autres apprentissages : elles développent l'esprit de synthèse, permettent le tâtonnement et la recherche de solutions par des chemins différents les uns des autres. L'informatique peut également favoriser les apprentissages chez certains élèves pour lesquels les méthodes classiques auraient échoué. Pour résumer, l'enseignement de l'informatique peut donc être utilisé soit comme un moyen pour former les élèves à d'autres disciplines ou comme un moyen d'atteindre le but général de l'école qui est de former des citoyens éclairés. Ceci bien évidemment sans exclusivité.

3) Comment enseigner la pensée informatique à l'école ?

La didactique de l'informatique est encore aujourd'hui très peu développée et l'informatique est encore perçue au travers de son expérience personnelle. Or, il ne s'agit pas ici de s'inscrire dans la position de l'utilisateur d'un ordinateur comme on a pu le faire jusqu'à aujourd'hui avec le B2i. L'informatique est un domaine complexe qui fait appel à de nombreuses compétences et qui est devenu un enjeu scientifique et sociétal. Pour enseigner l'informatique, il n'est pas nécessaire de tout savoir et de tout comprendre de l'informatique ; ne pas avoir suivi de cours d'informatique durant son cursus universitaire n'est pas un problème en soi. Il est essentiel de discerner :

- > Quel type de compétences sous-jacentes à la pensée informatique l'on se propose de travailler avec les élèves.
- > La conceptualisation (identifier, définir, dissocier...) de l'objet d'enseignement (raisonner donc en termes d'objectifs pédagogiques)
- > Les moyens utiles pour atteindre ces objectifs pédagogiques (typologie d'exercices, langage de programmation...)

Quelles notions aborder dans le cadre de cet enseignement ?

- ➔ Notion d'action / instruction : déplacer un personnage, effectuer un calcul...
- ➔ Structure conditionnelles : si...alors...sinon... / boucle (répéter l'action n fois)
- ➔ Mécanisme de lancement de script : quand on appuie sur un personnage, un drapeau...

- ➔ Aborder la notion de variables (plus difficile pour des élèves du CM1/CM2) : faisable en fonction du niveau des élèves.

Quel choix opérationnel mener à bien ?

Il serait dommage de ne placer l'enseignement de l'informatique que sous son angle algorithmique. Tout l'intérêt de cet enseignement passe par son côté « transdisciplinaire ». Il est en effet facile de définir des exercices mêlant l'informatique avec d'autres domaines ; d'autant plus que la plupart des apprenants baignent déjà dans un environnement numérique. Ces pratiques ouvrent une large place au travail collaboratif, à la créativité, à la méthodologie, à la résolution de problèmes et plus généralement à la capacité d'analyse. Comment cependant en pratique imbriquer ces attendus de manière opérationnelle ?

Travailler en informatique passe par l'utilisation d'un langage compréhensible à la fois par l'humain et aussi par la machine. Scratch a été développé dans ce sens tout particulièrement pour les élèves d'âge scolaire ; il a été pensé pour rester ludique, il est largement partagé par une vaste communauté et propose un environnement technologique stable, des structures algorithmiques de base et des mécanismes de synchronisation clairs. Son utilisation en ligne ou son installation sur les ordinateurs de l'école reste triviale.

4. Quels objectifs pédagogiques considérer ?

L'enseignement de l'informatique peut prendre des formes très différentes

	Objectif opérationnel	Remarque
1	Faire comprendre la notion d'algorithme et les notions sous-jacentes (action/instruction, séquence, boucle, structure conditionnelle, etc.)	A partir d'algorithmes simples de la vie quotidienne (recette de cuisine, multiplication d'entiers, accord du participe passé) et/ou des choses plus spécifiques (tri, recherche dichotomique, etc.)
2	Construire / élaborer des algorithmes	Complexifier progressivement
3	Utiliser un langage	Compréhensible par l'élève et la machine (scratch)
4	Pratiquer la programmation	Utiliser scratch, un robot (déplacements pour commencer)
5	Diversifier les usages du langage de programmation	Créer un environnement
6	Modéliser Décomposer des tâches complexes en tâches simples	Réaliser un projet mené sur plusieurs séquences, avec des avancées plus ou moins prononcées selon l'appropriation des élèves
7	Utiliser des objets pour enrichir l'environnement	Calculs, photos, sons capturer aux moyens de techniques apprises auparavant
8	Questions d'éthique, de principes	Utilisation et collecte des données. Droits et devoirs
9	Introduction progressive de nouvelles fonctions de programmation	Créer une didactique de l'informatique, travailler la logique, opérations conditionnelles de plus en plus complexes...
10	Utiliser contexte ludique riche et moteur de créativité	Souci prégnant de l'approche interdisciplinaire (programmation créative)
11	Avoir le souci que <u>tous</u> les élèves développent leur créativité	Faire participer et progresser tous les élèves
12	Permettre l'expression de la créativité	Favoriser les échanges, la prise de parole, les difficultés rencontrées : le langage oral

Mener une activité de programmation, ce n'est pas :

- Faire utiliser un ordinateur et des logiciels de bureautique (c'est l'objet du B2i)
- Faire utiliser un programme Scratch (tout fait) à des élèves
- Faire des « informaticiens » de nos élèves
- Faire programmer aux élèves un jeu ludique

Quelles approches de la pensée informatique adopter ?

→ Soit une approche orientée algorithmique

On se positionne dans ce cas dans une approche « informatique de la discipline ». L'idée centrale est qu'il faut enseigner l'algorithmique et ses notions sous-jacentes, ce qui nécessite de savoir faire des algorithmes et de savoir utiliser un langage de programmation.

→ Soit une approche orientée créativité.

C'est ce qu'appellent les américains : the four C's :

Critical thinking and problem solving, Communication, Collaboration and Creativity and innovation. Il s'agit de permettre à l'élève de développer sa créativité par la pratique informatique. C'est « l'informatique créative ». Plutôt que de parler de faire faire des algorithmes ou des programmes, il s'agira de produire des « médias interactifs » pour amener les élèves à développer un projet personnel. A ce titre, Scratch, qui fait l'unanimité, a été conçu dans une vision constructiviste des apprentissages (Seymour Papert / Jean Piaget). C'est une théorie de l'apprentissage qui reprend les principes constructivistes et développe l'idée que l'engagement de ces derniers dans des projets qui font sens pour eux est un facteur particulièrement favorable aux apprentissages

Ce type d'activités peut tout à fait s'inscrire dans une pratique de classe (longue) inscrite dans une démarche de projet qui fait appel à la réflexion, à l'échange, à la modélisation mais aussi au tâtonnement, à l'essai/erreur. Un cahier de « conception » peut alors être ouvert en ce sens sur lequel figureront les objectifs à atteindre, clarifiant les tâches à réaliser avec éventuellement des priorités. Ces tâches, distinctes, pourront être attribuées aux élèves par binôme au travers d'activités langagières et dans le cadre d'un travail collaboratif. Il pourra être présenté sous forme de tableau, facilement compréhensible par l'ensemble des élèves, témoin également de l'avancée du projet par un éventuel jeu des couleurs.

5. Utiliser Scratch pour enseigner la pensée informatique.

Scratch est un langage de programmation ; il permet de réaliser des tâches (déplacements, calculs,...).

Comme préalable, il s'agira de définir :

- Le ou les objectifs à atteindre
- La tâche à réaliser pour les élèves permettant de réaliser une activité cognitive (représentation mentale, comportement adapté,...)
- L'environnement pédagogique permettant d'atteindre le ou les objectifs fixés.

Si l'enseignant inscrit sa pratique dans une démarche de projet, il peut alors amener ses élèves à produire un scénario (problème complexe en langage ordinaire) qu'il s'agira alors de traduire en langage informatique (réalisation de l'algorithme, décomposition du problème complexe en problèmes simples, apprentissage du langage Scratch, essais/erreur, résolution de problèmes...).

Ce scénario pourra tout à fait s'inscrire dans le cadre d'une production d'écrit classique (dialogue, conte...) qu'il s'agira de coder (dialogues, déplacement des personnages, lieux...). Même si ce travail peut à priori paraître ludique, il va faire appel à des compétences cognitives élevées : définition du cahier des charges (que doit-on réaliser ? avec quoi ? en combien de groupes ? Sur combien de temps), du schéma narratif, de l'environnement graphique permis par Scratch (description), écriture des dialogues, respect de la syntaxe et des formes orthographiques, traduction du langage « réel » en langage « machine » grâce au logiciel Scratch qui sert d'interface de programmation. Analyse du script (traduction mentale de ce que chaque ligne de script va opérer). Amélioration possible du projet par ajout de scripts. Il permet en outre d'engager des débats entre les apprenants de la classe / du groupe : il permet d'exercer la pensée et les langages (français ou anglais écrits ou parlés, langage de programmation), avec et sans machine. Chaque séquence du projet pourra être alors considérée comme la réalisation d'un module (sous-programme) qui viendra s'insérer dans l'algorithme définitif (programme définitif abouti). Il s'agira d'amener l'élève à réfléchir sur l'effet opéré par le changement d'un paramètre (que se passe-t-il si l'on répète la boucle 18 fois au lieu de 20 fois ?). Comme dans la démarche d'investigation, on s'efforcera de

ne changer qu'un seul paramètre à la fois et l'avancée du projet passera nécessairement par des phases de test, d'évaluation, d'adaptation et de modification, un peu comme le font les informaticiens professionnels.

6) Quelle maîtrise l'enseignant doit-il avoir de Scratch ?

Quelles que soient nos compétences en informatique, l'enseignement de la pensée informatique avec Scratch reste toujours possible. Il ne s'agit pas de se placer dans une position « d'expert » face à nos élèves. Beaucoup de ressources existent en ligne et intervenir sur une ligne de script (du programme en langage Scratch) nous permet intuitivement d'en comprendre le fonctionnement. Certains pourront parfois même être réutilisés et adaptés pour un autre programme. Consulter les documents existants permettra de développer d'autres compétences chez nos élèves, notamment en lecture. Bien entendu, on ne pourra se passer d'un temps d'apprentissage et d'adaptation plus ou moins grand selon les utilisateurs. Il est à ce titre vivement conseillé d'utiliser les ressources suivantes :

Synthèse d'utilisation : [Scratchfr.free.fr/j8y3r7/pc1.4fr070109A4.pdf](https://scratchfr.free.fr/j8y3r7/pc1.4fr070109A4.pdf)

Guide de référence : <http://scratchfr.free.fr/k1n8g7/ScratchRefGuidefrv14A4.pdf>

7) Comment définir et gérer des situations pédagogiques ?

Face à la diversité des moyens matériels, aux profils et aux niveaux souvent très différents de nos élèves, il est difficile d'apporter une réponse « clé en main ». Des supports existent cependant et pourront être utilisés à bon escient :

« Bien commencer avec Scratch, introduction à l'informatique » (<https://pixees.fr/?p=3372>) propose des activités éclectiques pouvant être abordées progressivement.

« L'informatique créative » insiste plutôt sur le développement de la créativité de nos élèves au travers des activités proposées dans l'enseignement de la pensée informatique. Elle peut s'exprimer dans la forme avec la catégorie « apparence » et dans le fond « notion de boucle ». Ce second support s'intègre parfaitement à tous les projets interdisciplinaires et fait appel à des compétences très différentes.

L'enseignant aura avant tout à construire des situations pédagogiques à partir d'objectifs d'apprentissage précis et de ressources déjà existantes et sur lesquelles s'appuyer. La place de la motivation et de l'implication de l'élève ne sera jamais à négliger.

8. Comment déjouer les difficultés ?

Il s'agit d'engager nos élèves dans une dynamique marquée par :

- La pratiquer d'activités de programmation régulières et progressives.
- Des conditions d'enseignement sereines, avec un environnement pédagogique stable et parfaitement défini.
- Des prérequis (comme la maîtrise de l'utilisation de la souris par exemple) qui auront été travaillés.
- Des situations de classe envisagées sous forme de problématique (problème à résoudre et pas forcément en mathématiques !)
- Changer l'image de l'erreur en introduisant la notion d'essai / erreur qui nous permet de corriger les lignes de script du programme.
- D'engager enfin chez nos élèves, un retour réflexif des apprentissages. En d'autres termes, quelle place laisse-t-on à la métacognition ?

Pour conclure, pratiquer, c'est s'approprier le langage Scratch ; communiquer, c'est s'engager dans une démarche réflexive qui nous permet de progresser en même temps que nos apprenants.