# Introduction

Absente du quotidien il y a 60 ans, ***l’informatique*** est aujourd’hui présente dans la poche d’une majorité d’entre nous. Malgré cette ***omniprésence***, sa compréhension est souvent vue comme quelque chose de réservé à une frange réduite de la population : « les informaticiens ». En découlent un certain fatalisme face aux problèmes rencontrés – les fameux bugs – mais surtout une difficulté à comprendre les changement actuels et futurs provoqués par la diffusion de l’informatique dans tous les domaines de l’activité humaine.

L’éducation à l’informatique apparaît donc clairement comme une nécessité. Cette nécessité est retranscrite dans le socle commun de compétences de 2016 sous la forme suivante :« *Il 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.* »

De prime abord, la mise en œuvre concrète de ces enseignements pourrait être imaginée comme impliquant systématiquement l’usage d’un ordinateur, voire d’une tablette. Cependant, tout comme monter une pièce de théâtre ne se résume pas à monter sur scène et improviser, faire de l’informatique est loin de se résumer à écrire du code derrière un écran.

Par ailleurs, cette vision des choses est aussi trop précipitée, car elle néglige 2 paramètres importants :

* le niveau d’équipement en matériel informatique des écoles qui est très hétérogène
* le degré de familiarité des élèves vis à vis de ce matériel qui peut être nul comme très avancé.

Pour contourner les écueils énoncés précédemment, il semble intéressant de mettre en œuvre les activités informatiques dites « *débranchées*», c’est à dire ne nécessitant pas l’utilisation, par les élèves, de matériel informatique.

Ainsi, la séquence ici proposée cherchera à répondre à la problématique suivante :

*C****omment utiliser les activités «  débranchées » afin de favoriser la construction et l’apprentissage des fondements de l’informatique chez les élèves de CM1 ?***

Pour cela, le présent dossier est structuré comme ceci:

**Mettre ici la table des matières**

1 – introduction

2, 3, 4 – fondements scientifiques

5 – Rapport aux programmes & contenu de la séquence

6, 7, 8 -

# Fondements scientifiques

Pour clarifier l’objectif de cette séquence, commençons tout d’abord par expliciter ce qui est entendu par « informatique ».

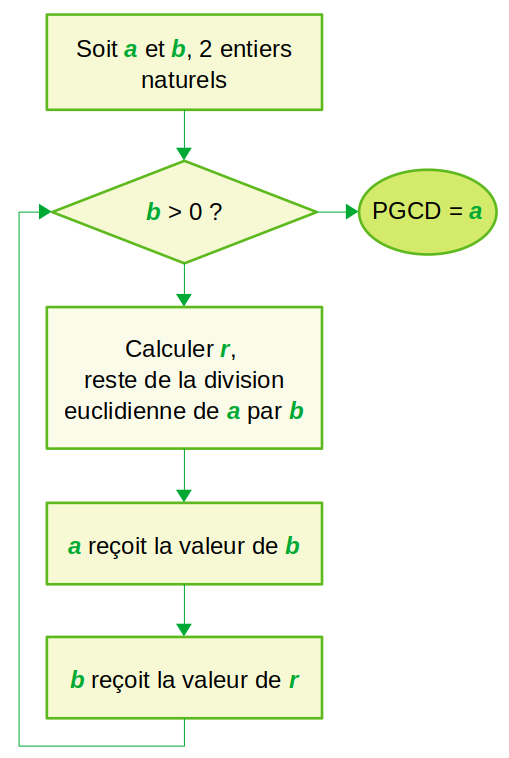
Le mot « Informatique », forgé par Philippe Dreyfus en 1962, peut être défini comme *la science et l’ensemble des techniques qui s’occupent de l’information sous toutes ses formes, de sa collecte, de son stockage et de son utilisation pour établir des programmes d’action*[[1]](#endnote-2). Ceci peut être résumé en « Science de traitement de l’information ».

Cette science repose sur 4 piliers, ou domaines d’étude[[2]](#endnote-3) :

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithmes | Machines | Langages & Programmes | Données |

### Les Algorithmes

Si l’on adopte une approche historique de l’informatique, les *algorithmes*, en tant que « *suite finie et non ambiguë d’opérations permettant d’obtenir un résultat* » en constituent assurément le plus ancien pilier. En effet, ceux-ci existent depuis que l’homme réalise des actions reproductibles dans un but donné : allumer un feu, tailler un silex pour produire un biface, etc.

  
Figure 1: *A*lgorithme *de calcul du* PGCD

Historiquement, l’un des premiers algorithmes a avoir été formalisé et à être arrivé jusqu’à nous est celui d’Euclide, qui permet le calcul du PGCD de deux entiers, et qui daterait de 300 ans avant JC[[3]](#endnote-4).

Au IXe siècle, le mathématicien perse Al-Khwârizmî écrit un traité de référence proposant une résolution systématique des équations du premier et second degré au moyen d’algorithmes. Cet ouvrage est une telle référence qu’il est considéré comme fondateur de l’algèbre et que son nom est l’origine du mot *Algorithme*.

De nos jours, des algorithmes sont créés dès lors qu’une activité humaine commence à faire l’usage d’outils informatique. Il existe cependant des familles d’algorithmes qui sont des « classiques » de l’informatique, de part les problèmes très répandus qu’ils adressent :

* Les algorithmes de tri, dont l’objectif est de classer les éléments d’une collection suivant une relation d’ordre souhaitée (tri par insertion, par sélection, le tri rapide, …).
* Les algorithmes de chiffrement, dont l’objectif est d’échanger des informations de manière confidentielle ou de s’authentifier à distance (chiffrement symétriques, chiffrement asymétriques, hachage, ...).
* Les algorithmes parcours de graphes, dont l’objectif est de permettre d’adresser des problèmes impliquant des réseaux reliant des nœuds (recherche de plus court chemin, recherche d’arbre couvrant, ...)

Généralement, pour répondre à un besoin, il n’existe pas une seule réponse algorithmique mais plusieurs. Aussi, un algorithme sera considéré comme étant le plus adapté quand, suivant les paramètres initiaux et les contraintes imposées, le temps nécessaire à son exécution (sa complexité temporelle) et de la quantité de mémoire qu’il va mobiliser (sa complexité spatiale) seront les plus réduits.

Un algorithme devra aussi être « correct », c’est à dire qu’il devra :

* se terminer (avoir une fin avec les entrées fournies)
* fournir un résultat conforme à la spécification (la description du besoin espéré)

Enfin, il est important de signaler qu’un algorithme, en tant que formalisation de la pensée humaine, ne pourra être considéré comme neutre[[4]](#endnote-5). Lorsqu’un s’agit de commander un robot chargé de peindre des carrosseries de voitures, cela est négligeable. Quand l’algorithme doit aider un juge dans sa prise de décision, la mise en œuvre de mesures permettant d’assurer son impartialité devra faire partie des critères d’évaluation de sa « correction ».

### Machines

Si la vocation première des algorithmes a été de formaliser un savoir dans le but d’en faciliter sa réutilisation ou sa transmission (recette de cuisine, algorithme mathématique), l’idée d’en déléguer l’exécution à une machine « programmable » a pour la première fois été mise en application avec le métier à tisser de Joseph Marie Jacquard, au début du XIXe siècle. L’algorithme, représenté sous la forme de trous dans des cartes perforées, commandait à la machine la relève de crochets, automatisant une opération jusqu’à présent manuelle.

Si la machine est ainsi rendue « programmable », elle ne peut pour autant pas être considérée comme un outil informatique, car elle ne manipule pas de données.

La manipulation de données apparaît à la fin du XIXe siècle, avec les premières tabulatrices. Celles-ci, créées pour permettre le recensement d’une population américaine[[5]](#endnote-6) devenue trop nombreuse pour un recensement « manuel », manipulent des cartes perforées servant au stockage de l’information. Elles sont capable de réaliser différents tris, et leur usage est rapidement élargi à d’autres domaines comme l’assurance ou le transport.

Elles sont cependant limitées à des activités de tri et ne permettent pas encore de constituer le « Graal » de machine universelle conceptualisée par Allan Turing en 1936.

Une machine de ce type doit être capable de traiter n’importe quel algorithme, du moment qu’elle peut gérer les volumes de données associés.

Les recherches et réalisations sur le sujet vont être favorisées par la seconde guerre mondiale, et en 1945 sera mis en service l’ENIAC, considéré comme le premier ordinateur.

Au fil des années, des progrès sont réalisés sur la puissance de calcul, la capacité de stockage ou encore la taille de ces appareils, pour aboutir à la situation actuelle où l’on trouve un avatar d’ordinateur dans tout dispositif dit « connecté ».

La notion de « machines » dans le domaine informatique mérite cependant d’être élargie par l’extension dont il est maintenant quasi indissociable : Internet.

Initié dans les années 1960 sous le nom d’ARPANET, sa finalité était de faire communiquer les ordinateurs distants de quelques universités américaines dans le cadre de recherches en informatique à destination de l’armée. Au fil des ans, le nombre d’ordinateurs connectés augmenta, les applications de courrier électroniques et de forum se répandirent, et de nouveaux protocoles (TCP/IP notamment) firent d’Internet la « surcouche » permettant de connecter entre eux l’ensemble des réseaux existants.

L’invention du Web au CERN en 1989, construit sur le concept des liens hypertextes, allait ouvrir Internet au grand public et permettre la création des services de plus en plus variés : moteurs de recherche, page web, blog, vidéos en ligne, réseau social, … Ces services étant eux même l’implémentation de nombreux algorithmes sous forme de programmes informatiques, écrits dans divers langages.

### Programmes & Langages

Entre l’algorithme et la machine, il a fallu imaginer un intermédiaire permettant de transformer une « idée » en des ordres entraînant des modifications d’état sur les machines.

Remontons encore une fois au métier à tisser de Jacquard. Celui-ci a grandement simplifié la réalisation de motifs complexes grâce à l’emploi de cartes perforées qui commandaient la relève ou non des fils sous lesquels devait passer la « navette ».

La notion de « Programme » apparaît de manière assez claire : il s’agit de l’ensemble des instructions qui commandent le relève ou non des fils, soit le lot de cartes perforées utilisé pour la réalisation d’un lai de tissus.

La notion de « Langage » est elle aussi assez évidente : soit il y avait un trou sur la carte, et le fil était relevé, soit il n’y en avait pas et le fil restait en place. On disposait donc d’un langage à 2 mots : « oui » et « non ». Ce langage ne se limitait pas à ces 2 mots. Il disposait aussi d’une syntaxe spécifique voulant que :

* sur une carte était présent un nombre fixe d’emplacements correspondant exactement au nombre de fils à commander (on peut considérer qu’une carte représentait une « phrase » du langage),
* chaque carte commandait la levée des fils pour un et un seul passage de la navette (on est ici sur le « sens » des « phrases » que l’on a construit, dont découlent les effets de celle-ci à l’exécution du programme, c’est à dire la « Sémantique »)

Vocabulaire, syntaxe, sémantique : les bases posées sont celles qui seront utilisées pendant près d’un siècle et demi, tant dans les tabulatrices que dans les automates de plus en plus évolués.

Autour de 1950, arrivent les premiers ordinateurs. Les instructions leurs sont passées au moyen d’un « langage machine », composé d’instructions directement compréhensibles par l’unité logique. Ce langage s’écrit tout d’abord au moyen d’interrupteurs qui sont ouverts ou fermés, puis de code sur des cartes perforées. Si la formalisation des algorithmes n’a plus que pour seule limite la capacité de ces machines, l’écriture de programmes exige un tel niveau de connaissances qu’elle reste le fait de rares experts.

Plusieurs personnes, dont l’américaine Grace Hopper, travaillent sur ce sujet pour aboutir, au début des années 60, à la création des langages de programmation dits « de haut niveau ». Leur principe est simple : rendre l’écriture de programmes plus efficace en utilisant une syntaxe plus proche de l’anglais. Sans être des langages naturels, ils sont plus souples que le langage machine, car ils proposent nativement vocabulaire plus riche intégrant les notions de conditionnelles, de boucles, l’utilisation de variables, de procédures ou encore de commentaires. Ils présentent aussi l’avantage de rendre les programmes moins dépendants d’une machine, car ils laissent à un « compilateur » la charge de retraduire en langage machine les instructions[[6]](#endnote-7).

De nos jours, si la programmation « bas niveau » reste utilisée sur les machines à très faibles capacités, les possibilités offertes par les langages de haut niveau se sont décuplées, avec d’une part des langages de plus en plus riches, et d’autre part, une pratique très répandue de partage et de réutilisation de briques déjà existantes, donnant parfois à la pratique de l’informatique des airs de jeu de construction.

### Les Données

Si, comme vu précédemment, les algorithmes peuvent s’appliquer à des situations et des objets « du monde réel », dans le cadre de l’informatique et des machines qui la sous-tendent, ceux-ci ne pourront s’appliquer qu’à des « données ». Par « données », il faut entendre « représentation de l’information, telle qu’elle est stockée ou exploitée dans les systèmes informatiques[[7]](#endnote-8). ». Cette « représentation » de l’information, forcément partielle, sera différente suivant la finalité de l’algorithme. Si l’on prend l’exemple d’une personne physique : dans un programme de recensement de population, les données gérées seront le nom complet, la date de naissance, l’adresse, etc, alors que dans un programme de type « réseau social », le pseudonyme, les relations ou les publications seront plus importantes que les données d’identification.

Les capacités des premiers ordinateurs étant limitées, le stockage des données n’était pas un problème majeur. Cependant, dans le contexte du programme Apollo, les capacités des machines et le volume de données augmentant, les premiers programmes dédiés, appelés Systèmes de Gestion de Bases de Données, furent créés dans le but de structurer et rendre efficace la gestion de la donnée sur ces machines. D’abord basés sur des modèles arborescents, ils évoluèrent ensuite vers le modèle « relationnel », visant à garantir cohérence et intégrité de l’information, puis vers le modèle orienté document, proposant une disponibilité de l’information dans le contexte d’Internet et de ses milliards d’utilisateurs.

En parallèle de cette gestion de l’information sous forme de données « textuelles », les capacités des machines dans les domaines du son et de l’image ont aussi nécessité d’imaginer des modes de gestion d’informations de nouveaux types : images, vidéos, sons, odeurs... tout ceci à base de 0 et de 1 : on parle ici des formats.

Enfin, l’étude de la sécurisation des données, qui existait avant l’informatique (chiffre de César, bâton de Plutarque,…) a fait et fait toujours l’objet de travaux très importants, à la croisée de l’informatique et des mathématiques.

# L’enseignement de l’informatique dans le cadre des sciences et technologies

### Historique et évolutions

L’enseignement de l’informatique à l’école primaire française a connu une première apogée au milieu des années 1980, lors du plan « Informatique Pour Tous ». À une époque où les possibilités offertes par les ordinateurs étaient encore limitées, le parti pris avait été d’enseigner les bases la programmation. Les élèves étaient mis en situation de création via des activités comme la programmation des déplacements d’une tortue virtuelle. Relégué quelques années plus tard au second rang des priorités, l’informatique revint sur le devant de la scène au début des années 2000. Entre temps, les usages ont explosés (bureautique, photo numérique, internet,…), et l’attention des programmes est portée sur le savoir-utiliser plutôt que sur le savoir-faire[[8]](#endnote-9). Avec la réforme des programmes de 2016, les fondements de l’informatique sont de retour (cf introduction).

### Informatique débranchée : intérêt pédagogique

Outre le fait que pratiquer l’informatique à l’école se heurte parfois à des contraintes logistique de disponibilité de matériel, la pratique de l’informatique débranchée présente de nombreux intérêts pédagogiques[[9]](#endnote-10).

Elle permet tout d’abord de démystifier l’informatique, de la rendre concrète, en la raccrochant à des activités tangibles du monde réel, proche des représentations des élèves. Elle permet ensuite une prise de conscience des élèves sur le fait que le cœur de l’informatique est la réflexion, et non pas la manipulation d’un objet électronique.

L’obsolescence rapide de ces objets électroniques induit par ailleurs une impression de péremption rapide des savoirs liés à informatique. A l’inverse, le détachement des effets de mode proposé par l’informatique débranchée donne à voir la constance dans les savoirs à acquérir[[10]](#endnote-11).

Elle permet la découverte d’une activité en laissant initialement de côté la charge cognitive liée à la découverte d’un outil comme Scratch.

Elle permet enfin de ne pas rajouter du temps d’écran supplémentaire aux enfants, à l’heure où celui-ci est considéré comme délétère par certains chercheurs.

# Séquence proposée

### **Présentation générale**

De part son contenu, cette séquence a vocation à servir de porte d’entrée aux élèves dans le monde de la pensée informatique. Elle sera donc positionnée en début de cycle 3 et ne nécessitera pas de pré requis spécifiques (elle ne sera donc pas l’objet d’une évaluation diagnostique). Pour les élèves ayant déjà abordé ces sujets en cycle 1 ou 2, elle permettra une réinvestissement et un approfondissement des compétences précédemment travaillées.

Si les pages précédentes ont détaillé l’histoire et les fondements de l’informatique en tant que science, pour des élèves de cycle 3, la difficulté du sujet et les capacités importantes d’abstraction qu’il demande conduit à privilégier une approche du sujet sous l’angle technologique.

Ainsi, les séances proposées vont s’attacher à suivre une démarche habituellement utilisée pour la fabrication d’un objet technique : analyse du besoin, spécifications, réalisation, tests, corrections éventuelles, conclusion.

Cette séquence est construite en 5 séances : les 4 premières servent à l’acquisition des premières notions associées à chacun des « piliers » de l’informatique, la cinquième sert à un réinvestissement/synthèse et à une évaluation.

Pour la construire, les 2 principales sources sont les activités proposées par l’IREM de Grenoble et l’INRIA, ainsi que les activités « *1-2-3 codez »* de la Fondation La main à la pâte.

### Compétences travaillées

L’utilisation systématique d’activités débranchées, travaillées sous forme d’activités de groupe, vont permettre de travailler les compétences suivantes du socle commun :

|  |  |
| --- | --- |
| Domaine | Compétence |
| Domaine 1 | Comprendre, s’exprimer en utilisant la langue française à l’oral et à l’écrit |
| Domaine 1 | Pratiquer les langages scientifiques |
| Domaine 3 | La formation de la personne et du citoyen (développer la confiance en soi et le respect des autre) |
| Domaine 4 | Pratiquer des démarches technologiques |
| Domaine 4 | Concevoir, créer, réaliser |

Plus spécifiquement sur les attendus de fin de cycle en Sciences et Technologies, cette séquence permettra de travailler **:**

Thème (1) Matière, mouvement, énergie, information

* Identifier un signal et une information (séance « données »)

Thème (3) Matériaux et objets techniques

* Décrire le fonctionnement d’objets techniques, leurs fonctions et leurs constitutions (toutes séances)
* Concevoir et produire tout ou partie d’un objet technique en équipe pour traduire une solution technologique répondant à un besoin (toutes séances)
* Repérer et comprendre la communication et la gestion de l’information (toutes séances)

### **Séance 1 : A la découverte des algorithmes - 1h**

|  |
| --- |
| **Objectifs séance :**   * Découvrir les 4 piliers de l’informatique * Découvrir plus précisément la notion d’algorithme * Rechercher un algorithme permettant de répondre à une problématique donnée   *Cette séance est basée sur l’activité proposée par Marie DUFLOT – INRIA dans le cadre de CLASS CODE.* |
| **1. Introduction de la séquence par la découverte des 4 piliers de l’informatique** |
| |  5 min | Discussion |  : Vidéoprojecteur / Photo de Bill GATES / Tableau de la classe |
| Écriture du mot « Informatique » au centre du tableau.  Questionnement des élèves sur ce qu’est pour eux l’informatique. Écriture des réponses au tableau, classées en 4 zones distinctes.  Une fois toutes les réponses notées, présentation des 4 regroupements : algorithmes, machines, programmes & langages, données, comme étant les 4 piliers de l’informatique.  Précision de ce qui se cache derrière chacun des 4 piliers, et annonce du contenu de la séquence à venir en 5 séances. |
| **2. Introduction du sujet principal de la séance : la notion d’algorithme** |
| |  5 min | Discussion |  : Tableau de la classe |
| Présentation du concept d’« algorithmes » : suite finie et précise d’opérations permettant d’obtenir un résultat donné à partir d’une situation de départ.  Questionnement des élèves sur les algorithmes de la vie de tous les jours qu’ils connaissent ? Compléments éventuels apportés : manuel d’instruction de Lego, collier de perles à reproduire en petite section, recette de cuisine. |

|  |
| --- |
| **3. Présentation du cas du « crêpier psychorigide »** |
| |  5 min | Consignes |  : Grandes « crêpes » de tailles différentes (visibles fond de classe) |
| Présentation de la situation : « *Comment un crêpier peut-il faire pour classer ses crêpes de la plus grande à la plus petite, sachant qu’elles forment une pile unique et que la seule manipulation autorisée consiste à glisser sa spatule sous une crêpe de la pile et à retourner toutes les crêpes dessus celle-ci.* »  Démonstration aux élèves (sans détailler la méthode) de la résolution d’un exemple.  Consigne aux élèves : déterminez l’algorithme permettant d’apporter une solution au crêpier, quel que soit le nombre de crêpes et leur ordre. Vous pouvez expérimenter avec les « crêpes » à votre disposition. Suivant vos résultats, un ou plusieurs groupes viendront devant la classe. Testez bien votre solution, chacun dans le groupe doit la maîtriser. Cette solution doit prendre la forme de « consignes » à donner à quelqu’un effectuant les manipulations de tri à votre place.  Éventuellement : définir « psychorigide » et répéter la définition d’algorithme. |
| **4. Recherche d’un algorithme** |
| | 15 min | Recherche |  : Lot de 5 « crêpes » pour chaque groupe + consignes au tableau |
| Dans un premier temps, encourager les élèves à expérimenter.  Circuler dans la classe afin d’observer le fonctionnement de chaque groupe. Répondre aux questions. Aider ceux qui semblent bloqués (questionnement, aide à la verbalisation, suggestion de « récursivité » quand la crêpe la plus grosse a été mise sur le dessous). Prise en photo des mains des élèves en train de manipuler. Aux groupes pensant avoir réussi, suggestion de tester en partant d’une pile initiale ordonnée différemment et de bien valider leur algorithme (tests). Leur proposer de tester leur solution sans voir la pile, afin de s’assurer de la clarté de l’expression de l’algorithme.  Différentiation 1 : Si au bout de 10 minutes, la majorité des groupes semble loin de la solution, suggérer d’essayer de trouver une méthode pour mettre la plus grande crêpe en bas de la pile.  Différentiation 2 : Une fois la solution trouvée dans des groupes, inciter ceux ci à perfectionner leur solution afin d’en retirer les étapes inutiles.  Différentiation 3 : Si certains groupes trouvent rapidement la solution, leur proposer de déterminer l’algorithme permettant de ranger les crêpes en aillant la face la plus cuite vers le bas. Ceci permettra d’introduire la notion de condition. |

|  |
| --- |
| **5. Test des algorithmes conçus par les groupes** |
| |  20 min | Mise en commun |  : Grandes « crêpes » |
| Il sera intéressant de faire venir devant la classe le ou les groupes n’ayant pas réussi, ceci dans le but que la classe les aide à trouver une solution.  Le porte-parole qu’il se désigne va me donner les consignes que je vais exécuter avec le jeux de « grandes » crêpes, devant l’ensemble de la classe.  Les consignes erronées seront l’opportunité d’illustrer le fait que la machine ne sait rien d’autre que ce qu’on lui demande, ainsi que le concept de bug.  Les consignes incomplètes permettront d’insister sur leur nécessaire précision pour que la machine soit capable de les exécuter.  Arrivé au point de blocage du groupe, je questionne les autres groupes afin qu’ils aident à identifier une solution fonctionnelle.  Une fois que la classe a trouvé un algorithme fonctionnel, il sera appliqué à plusieurs situations de départ différentes afin d’en montrer la généricité (en série puis en parallèle). |

|  |
| --- |
| **6. Rédaction de la trace écrite dans le cahier de science** |
| |  5 min | Institutionnalisation |  : Cahier de sciences |
| Écriture par les élèves des 4 piliers de l’informatique, accompagnés de 1 ou 2 exemples caractéristiques.  Écriture de la définition d’un algorithme en face du pilier « algorithme ». Un espace est laissé à côté des 3 autres afin que la définition soit ajoutée lors des séances suivantes.  Collage dans le cahier de science d’une représentation schématique de l’algorithme de tri des crêpes, accompagnée d’une description textuelle de celui-ci.  Conclusion en rappelant aux élèves qu’ils ont fait de l’informatique… sans ordinateur.  Collage (ultérieur) de la photo des mains de chaque groupe en train de manipuler afin de conserver une trace de l’expérimentation pratiquée. |

### **Séance 2 : A la découverte des langages informatiques - 1h**

|  |
| --- |
| **Objectifs séance : découvrir quel langage utiliser pour communiquer avec une machine**   * Construire un langage utilisable pour diriger une machine * Préciser la notion de langage   *Cette séance est basée sur l’activité 1-2-3 Codez de la fondation la Main à la pâte.* |
| **1. Introduction : rappel du contexte de la séquence et présentation thématique langages** |
| |  10 min | Rappels et découverte |  : Vidéoprojecteur avec photos de Margaret Hamilton |
| Questionnement des élèves pour rappeler les 4 piliers de l’informatique vue en séance 1.  Annonce du thème des langages pour la séance du jour.    Affichage de la première photo et questionnement des élèves sur le nom et le métier de la personne.  Énoncé du nom de Margaret Hamilton. Affichage de la seconde photo et questionnement sur le métier.  Affichage de la troisième photo et réponse : Margaret Hamilton est une informaticienne qui a dirigée l’équipe en charge de concevoir les programmes informatiques de la mission Apollo qui a envoyé des hommes sur la lune en 1969. Utiliser la seconde photo pour insister sur le fait que l’informatique ne se fait pas nécessairement devant un ordinateur. |
| **2. Objectif Lune** |
| |  5 min | Découverte |  : Vidéoprojecteur / Image du parcours d’atterrissage |
| Mise en situation des élèves : vous êtes en 1965, Margaret Hamilton fait appel à votre classe pour l’aider dans la programmation du module lunaire (LEM) chargé de déposer les astronautes sur la lune.  Plus spécifiquement, vous allez devoir programmer la phase d’atterrissage sur la lune, en respectant le trajet permettant d’éviter les zones dangereuses.  Pour cela, il va vous falloir tout d’abord définir quel type d’instructions donner au LEM pour lui faire suivre exactement le parcours tracé sur la carte.  Les déplacements se font « carreau par carreau ». |
| **3. Guidage du LEM – Recherche en petits groupes** |
| |  15 min | Recherche |  : Une fiche par groupe avec le parcours attendu |
| Je circule dans la classe afin d’observer le fonctionnement de chaque groupe. J’aide par le questionnement ceux qui semblent bloqués. Je réponds aux questions. Aux groupes pensant avoir terminé, je leur propose de faire individuellement un test afin de valider leur proposition, ou de détecter des erreurs le cas échéant.  Différentiation 1 : Si certains groupes trouvent rapidement la solution, leur proposer de réfléchir à un second mode de guidage, afin qu’ils trouvent les 2 logiques courantes : absolue et relative.  Différentiation 2 : Une activité alternative peut consister à faire réfléchir ces groupes sur le moyen de simplifier au maximum leur langage en réduisant si possible le nombre de mot et la longueur de ceux-ci. |
| **4. Guidage du LEM – Test des résultats des groupes** |
| |  15 min | Mise en commun |  : Vidéoprojecteur / Image du parcours + LEM à déplacer |
| Un représentant de chaque groupe vient faire exécuter sa solution par un membre d’un autre groupe (afin d’éviter les consignes « implicites »).  Les propositions ambiguës ou non conformes aux règles énoncées seront écartées car ne pouvant pas être comprises par la machine.  Les propositions ne permettant pas de suivre le parcours ou d’atteindre la cible seront utilisées pour mettre en avant le concept de bug. J’attirerai l’attention des élèves sur le fait que toute erreur conduit à un résultat faux, aux conséquences potentiellement terribles. Ce sera l’occasion de présenter le concept très important de tests. |
| **5. Définition d’un langage informatique** |
| |  15 min | Institutionnalisation |  : Cahier de sciences |
| Sont couramment proposés 2 types de langages : les langages absolus, utilisant des consignes du type « nord/sud/est/ouest », et les langages relatifs, utilisant des consignes du type « avance, pivote gauche, pivote droite ».  Questionnement des élèves sur les différences entre ces 2 langages. L’objectif est de mettre en évidence des vocabulaires différents.  Questionnement des élèves sur les différences avec le français. Les questions sont orientées afin d’arriver à la conclusion qu’un langage informatique dispose d’un vocabulaire spécifique et précis, d’une syntaxe qui lui est propre et d’une grammaire simple.  Écriture dans le cahier de science :  Pour donner des instructions à une machine, on utilise un langage de programmation, compréhensible à la fois par la machine et par l'être humain (sur le schéma des 4 piliers de l’informatique, en face du pilier « langages »).  Un langage de programmation est différent d’une langue naturelle car il possède très peu de mots et de règles de grammaire et ne laisse place à aucune ambiguïté.  Il existe de nombreux langages de programmation, adaptés à différents usages.  Un bug est une erreur dans un programme. Un bug minime en apparence peut avoir des conséquences énormes.  Les tests sont là pour détecter les bugs et les corriger. |

### Séance 3 : A la découverte des données - 1h

|  |
| --- |
| **Objectifs spécifiques de la séance : le concept de données**      *Cette séance est basée sur l’activité ‘Télé-vision’ pour laquelle Maryline Althuser a obtenu le 1e prix au trophée Shannon 2017.* |
| **1.** |
| |  5 min | Rappels et découverte |  : Vidéoprojecteur avec photos de Margaret Hamilton |
|  |
| **2.** |
| |  5 min | Découverte |  : Vidéoprojecteur / Image du parcours d’atterrissage |
|  |
| **3.** |
| |  15 min | Recherche |  : Une fiche par groupe avec le parcours attendu |
|  |
| **4.** |
| |  15 min | Mise en commun |  : Vidéoprojecteur / Image du parcours + LEM à déplacer |
|  |
| **5.** |
| |  15 min | Institutionnalisation |  : Cahier de sciences |
| Écriture dans le cahier de science de la définition de données |

### **Séance 4 : A la découverte des machines - 1h**

|  |
| --- |
| **Objectifs spécifiques de la séance :**   * Découvrir l’histoire de l’ordinateur * Découvrir le concept de protocole de communication de l’information   *Cette séance est basée sur l’activité ‘Internet avec des post-it’ disponible sur le site de médiation PIXEE de l’INRIA.* |
| **1. Histoire des ordinateurs** |
| |  10 min | Rappels et découverte |  : Frise vidéo projetée |
| Rappel du contexte de la séquence : dernière séance, dont le sujet sera « les machines »  Frise chronologique découverte progressivement  Tailleurs de silex : algorithmes « débranché » (sans machine)  Métier à tisser Jacquard : machine, langage de programmation, programme, mais pas calcul  Machine analytique de Babbage : algorithme programmable, capacité à faire des calculs.  Ada Lovelace : Première programmeuse informatique, sur la machine analytique  Alan Turing : définition de l’ordinateur au sens de machine capable de traiter tout algorithme si l’on fait abstraction de ses capacités mémoire.  ENIAC : premier ordinateur au sens de Turing.  PC, consoles, smartphones : les ordinateurs actuels i.e. machines capables de stocker de l’information, d’effectuer des calculs, de recevoir des informations en entrée et d’en retourner en « sortie ». |
| **2. De l’importance des réseaux** |
| |  5 min | Découverte |  : - |
| Questionnement des élèves sur leurs usages des objets informatiques.  Dès qu’un usage comme « messagerie » ou « envoi de photos » sort, interroger la classe sur l’élément qui permet cet usage.  Réponse possible : Internet.  Questionnement des élèves sur ce qu’est concrètement Internet.  Recueil des réponses et précision du concept en le présentant comme un ensemble de machines type ordinateur, communiquant entre elles via un « réseau » physique (fils téléphoniques, antennes relais, câbles sous-marins,…) et utilisant des « protocoles » ou « méthodes d’échanges d’information ».  Représentation en simultané au tableau.  De par son importance aujourd’hui, le réseau est considéré comme l’extension des machines. |
| **3. Internet avec des post-it – Présentation de l’activité** |
| |  5 min | Découverte |  : Post-it pour chaque groupe |
| Division de la classe en 3 groupes de 8 élèves.  Présentation de l’activité : nous allons maintenant imaginer un moyen de faire circuler de l’information. Les règles sont les suivantes :  - le but est d’envoyer un mot à un camarade du groupe  - ce mot sera écrit sur un ou des post-it  - il est autorisé d’écrire toute information utile sur le post-it  - les post-it circuleront sous le bureau, et il est interdit de parler pendant les phases de transmission d’information.  Au sein de chaque groupe, imaginer comment permettre cette transmission d’informations.  Testez là : chaque élève devra envoyer un mot et être le destinataire d’un mot. |
| **4. Phase 1 / envoi simple de mots** |
| |  10 min | Recherche |  : Post-it par groupe |
| Le but est que les élèves se mettent d’accord sur l’  Je circule dans la  - phase 1 : le routage simple  - phase 2 : le découpage en paquets  - phase 3 : l’accusé de réception (nécessité d’écriture de l’émetteur)  - phase 4 : la confidentialité |
| **4.** |
| |  15 min | Mise en commun |  : Vidéoprojecteur / Image du parcours + LEM à déplacer |
|  |
| **5.** |
| |  15 min | Institutionnalisation |  : Cahier de sciences |
| Définir protocole  Généraliser sur le concept de protocoles à tous les niveaux des machines ?  Écriture dans le cahier de science de la définition de machine !  Collage de la frise chronologique « machines » dans le cahier de science  Collage des post-it dans le cahier  Pollution liée aux machines et à l’usage des réseaux |

### Séance 5 : réinvestissement et évaluation :

- Concept de récursivité : montrer une série d’image, dont le logo de la vache qui ri. Demander quelle image est « récursive »

Problématique élève: Comment enrichir le langage afin de le rendre plus performant

* introduire les « procédures »
  + répétition
  + boucles
  + conditions
* Réinvestissement seul
  + Guider le robot en utilisant obligatoirement le langage relatif

evaluation ??

* *l’élève doit comprendre ce qu’il a acquis, ce qu’il doit travailler…*
* *l’évaluation doit être un réel apprentissage*
* *Comment travailler avec cet évaluation, comment l’utiliser pour faire progresser l’élève*
* *voir les compétences listées en début de séquence*

# Conclusion

Réponse à la problématique & ouverture

* Scratch: découverte de la programmation à partir des éléments construits en mode débranché
* Anglais: acquisition d’une série de mots et expressions liés à l’exploration lunaire réutilisables dans d’autres circonstances (Moon, earth, travel,…)
* Arts plastiques: usage des outils informatiques de travail de l’image et de recherche d’information, au service de la pratique plastique
* Mathématiques: calculs
* Education musicale (ground control to major tom)
* Concernant la formation à la pensée informatique, il semble important de ne pas se contenter de littératie (cf le discours d’Eric Schmit devant le parlement ? Anglais)
* Concours castor [ici](https://www.castor-informatique.fr/)
* EMC & histoire: IBM et l’haulocauste
* EMC : la partialité des algorithmes

1. Maurice Nivat. Une très vielle science, l’informatique. Sciences et technologies de l’information et de

   la communication en milieu éducatif : Analyse de pratiques et enjeux didactiques., Oct 2011, pp.33-47. edutice-00676195 [↑](#endnote-ref-2)
2. Gérard Berry, Education à l’informatique, Cours au Collège de France, 6 février 2019 [↑](#endnote-ref-3)
3. https://fr.wikipedia.org/wiki/Algorithme\_d%27Euclide [↑](#endnote-ref-4)
4. Bertail, Bounie, Clémençon, Waelbroeck : Algorithmes : biais, discrimination et équité, Institut Mines-Télécoms, Février 2019. [↑](#endnote-ref-5)
5. IBM et l’holocauste, Edwin Black [↑](#endnote-ref-6)
6. IREM Clermont-Ferrand, Langages de programmation, support pédagogique à destination des enseignants [↑](#endnote-ref-7)
7. https://fr.wikipedia.org/wiki/Donn%C3%A9e\_(informatique) [↑](#endnote-ref-8)
8. Gérard Berry, Éducation à l’informatique en France, Cours au Collège de France, 6 février 2019 [↑](#endnote-ref-9)
9. Robert Cabane, inspecteur général honoraire, Tangente Éducation n°42 [↑](#endnote-ref-10)
10. E. Vandeput, Didactique de l’informatique, 1024, Bulletin de la société informatique de France, numéro 8, avril 2016 [↑](#endnote-ref-11)