

Algorithmique et programmation, un levier pour développer des compétences mathématiques

Document rédigé par

Yannick Danard, professeur au collège Clément Janequin - Avrillé

*Stéphane Percot, professeur de mathématiques au collège Haxo de La Roche-sur-Yon
et IAN de mathématiques de l'académie de Nantes.*

Sous la direction de

Véronique Bluteau-Davy, IA-IPR de mathématiques dans l'académie de Nantes

Avec la collaboration de

Yann Bertrand Professeur au lycée Aristide Briand - Saint Nazaire

Fabrice Foucher Professeur au lycée Jacques Prévert - Savenay

Emmanuel Malgras professeur au collège Pierre et Marie Curie - Le Pellerin

Grégory Maupu professeur au collège Milcendeau – Challans

Damien Rivière Professeur au collège Pierre Dubois - Laval

SOMMAIRE

CE TRAVAIL A RÉVEILLÉ L'INFORMATICIENNE QUI EST EN MOI !.....	3
INTRODUCTION.....	4
DES POSSIBILITÉS DE PRISE EN MAIN	4
EXEMPLES DE PREMIERS PAS.....	6
VERS DES PROGRAMMES PLUS ÉLABORÉS... ..	7
I) QUAND L'ALGORITHMIQUE DEBRANCHEE FACILITE LA PROGRAMMATION.....	7
II) TRAVAILLER LES ANGLES A L'AIDE DE SCRATCH	10
III) MARCHES ALEATOIRES.....	10
L'INITIALISATION.....	12
DES OBJECTIFS PAS TOUJOURS ATTEINTS	12
LES COMPÉTENCES.....	14
DES PERSPECTIVES POUR L'AN PROCHAIN	16
ANNEXE 1	17
ANNEXE 2	19
ANNEXE 3	21

CE TRAVAIL A RÉVEILLÉ L'INFORMATICIENNE QUI EST EN MOI !

Voici quelques témoignages d'élèves de 6^{ème} :

Ce travail sur Scratch m'a beaucoup plu car ça permet de faire des mathématiques tout en faisant quelque chose de sympa pour les élèves.

Ce travail était pas très difficile mais il faut quand même bien réfléchir pour mettre tout les éléments dans le bon ordre.

J'ai adoré scratch surtout le jeu vidéo.

C'était simple et intéressant mais certaines choses étaient plus dur.

Mais je trouve que le travail sur Scratch a été bien et des fois difficile mes c'est bien.

J'ai adoré c'était trop bien.

Il y a juste le travail pluriel singulier qui n'était pas facile mais sinon c'était bien car on faisait des math tout en s'amusant.

J'ai adoré.

C'était trop bien.
C'était difficile mais j'ai réussi.
Je fais chez moi je m'entraîne.

Le travail sur Scratch été assez bien, ça m'a beaucoup plu. Ce travail a réveillé l'informaticienne qui est en moi. Je sais maintenant que créer des jeux n'est pas facile.

C'est juste trop BIEN! On s'amuse bien et c'est facile! Bon des fois c'est assez compliqué mais ça va!

Le travail sur scratch est intéressant et génial ça nous apprend beaucoup de chose sur la programmation. On peut faire tant de chose que c'est génial, (on peut faire un jeu vidéo, une conjugaison d'un verbe et faire des calculs).

J'ai vraiment aimé le travail sur scratch, ça m'a beaucoup plu et je trouvais qu'il y avait un peu de difficulté ce qui était bien, j'ai surtout aimé faire le jeu.

J'ai trouvé le travail sur Scratch intéressant et génial.

J'ai bien aimé car il fallait conjuguer le verbe être et avoir. Mais j'ai tout aimé et dommage que ses fini.

Mais non, mais non, ce n'est pas fini...

INTRODUCTION

« Les enseignants doivent connaître et comprendre la culture des jeunes pour en partir et les amener aux savoirs académiques » Serge Tisseron, journée du numérique de l'académie de Nantes, 23 mars 2016.

Ce document a pour objectif de rendre compte de la réflexion conduite au cours de l'année scolaire 2015-2016 dans le cadre des travaux mutualisés et d'identifier les leviers pour développer les compétences mathématiques que l'on peut mettre en œuvre par le biais de l'algorithmique et de la programmation.

Les approches informatiques ont été faites en utilisant principalement le logiciel Scratch. Ce logiciel permet une prise en main très rapide pour un usage basique et autorise ensuite des approches plus élaborées pour travailler des compétences en profondeur.

DES POSSIBILITÉS DE PRISE EN MAIN

Il apparaît que toutes les modalités de prise en main par des élèves de collège vont s'appuyer sur trois aspects essentiels de Scratch :

- la prise en main très facile du logiciel ;
- la possibilité de créer et de voir très rapidement le résultat de ses propres créations ;
- le côté intrinsèquement ludique qui induit l'adhésion des élèves à tous les projets.

De ce point de vue, il convient – dès les premiers instants en classe – de laisser une liberté d'approche.

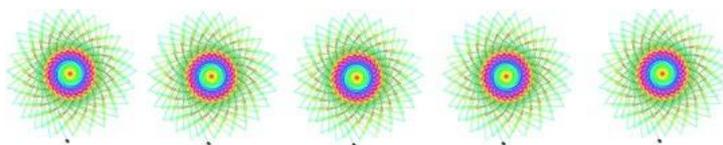
On peut s'appuyer sur deux entrées principales :

- les déplacements (la construction de figures géométriques peut intervenir dans ce cadre) : [quelques constructions géométriques](#) ; [programmes simples](#) ; [tableau de fil](#))
- la réalisation de calculs (programme de calculs, fonctions, compteurs, ...) [commandes au restaurant](#) ; [somme d'entiers consécutifs](#)

Ces deux entrées permettent d'aborder de façon simple et accessible les principaux éléments constitutifs de programmes plus élaborés tels que les jeux ou les tables de conjugaison par exemple. Par cette approche, les élèves sont acteurs de la réalisation des briques élémentaires qui constitueront la base de programmes plus ambitieux : on est ici sur la création de routines informatiques. Les jeux par exemple autoriseront une réelle différenciation à l'intérieur de la classe : un premier programme simple devra être accessible à tous, puis des évolutions vers une complexité accrue permettront à chacun de donner le meilleur de lui-même.

Cela n'empêche pas de laisser une liberté dans la création de formes géométriques classiques (carrés, rectangles, losanges, triangles). Cette liberté se verra d'abord essentiellement dans l'habillage : choix du lutin, couleur du trait, épaisseur du trait, ... tout en amenant peu à peu des éléments de programmation essentiels :

- la variable
- la boucle (répéter ... fois ; répéter jusqu'à ...)
- le test : si...alors... ; si ...alors ... sinon ...



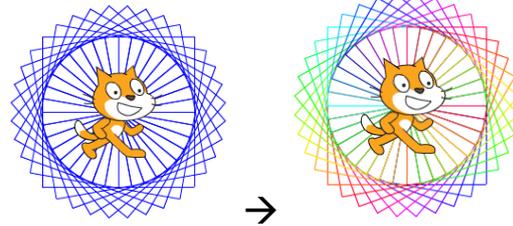
Voici par exemple quatre évolutions d'un programme :

1		<p>Dans ces deux programmes, on trace juste un carré :</p> <p>Ce programme permet d'aborder une première fois la boucle « répéter...fois ».</p>
2		<p>Le test « Si ... alors... » permet ici de ne pas tracer un carré trop grand, qui par exemple sortirait de la scène. Il a fallu de plus intégrer une variable.</p>
3		

4

```
quand flag pressé
s'orienter à 90
effacer tout
style en position d'écriture
demander "Quelle est la longueur du côté du carré ?" et attendre
mettre longueur à réponse
mettre compteur à 1
répéter jusqu'à compteur * 10 > 360
si longueur < 100 alors
  répéter 4 fois
  avancer de 80
  tourner de 90 degrés
  tourner de 10 degrés
  ajouter à compteur 1
relever le stylo
```

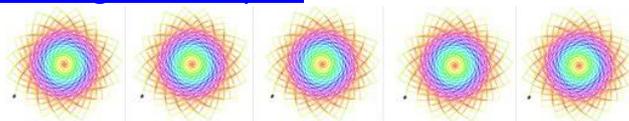
Ce dernier programme donnera un visuel très agréable qui plaira immédiatement aux élèves...surtout si on y ajoute des modifications de couleurs.



Et on aborde la boucle « répéter jusqu'à... »

Ces figures ne sont constituées que de carrés.

Lien : [Quelques constructions géométriques](#)



EXEMPLES DE PREMIERS PAS

Une difficulté qui peut être ressentie par un professeur est celle des premières minutes : comment démarrer concrètement ? L'objectif est de taille : il s'agit de mettre les élèves en situation d'aborder ce nouveau thème du programme en confiance. Il convient donc de jouer sur l'accessibilité et de s'appuyer sur les aspects ludiques et créatifs du logiciel.

1) Démarrer avec l'heure de code : <https://code.org/learn> pour une prise en main guidée et progressive de la programmation et l'assemblage par blocs

Il s'agit de déplacer un personnage avec des mouvements de plus en plus élaborés au fur et à mesure. Pour cela, on déplace et on assemble des blocs comme cela se fait dans Scratch.



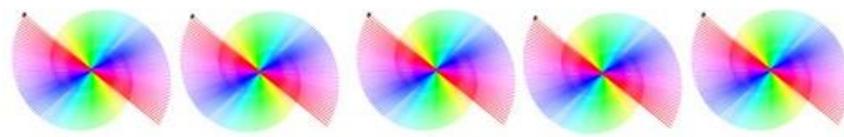
2) Démarrer en montrant pendant quelques minutes les éléments de base.

L'expérience montre qu'en classe de 6^{ème}/5^{ème} par exemple, une dizaine de minutes permet de mettre en évidence qu'on peut démarrer en appuyant sur le drapeau vert, qu'on peut déplacer le lutin et le faire écrire.



Ces 6 éléments suffisent pour démarrer.

Une quinzaine de minutes de totale liberté sur le logiciel pour essayer et tester suffit pour une prise en main basique du logiciel. On peut alors donner des objectifs de constructions à atteindre.



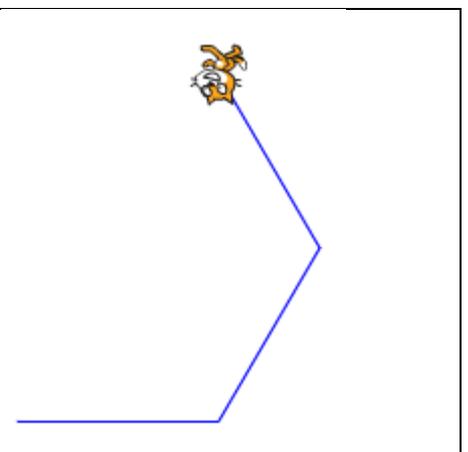
VERS DES PROGRAMMES PLUS ÉLABORÉS...

1) Quand l'algorithmique débranchée facilite la programmation

Une approche sous la forme d'algorithmique débranchée facilite souvent la compréhension des outils qu'on utilisera dans la programmation. C'est particulièrement vrai pour la variable mais on le vérifie aussi sur les boucles ou les tests conditionnels par exemple. Pour autant, il peut être nécessaire que l'élève soit confronté à la difficulté dans la réalisation d'un programme avec Scratch pour que l'activité débranchée prenne tout son sens.

1) Le triangle équilatéral

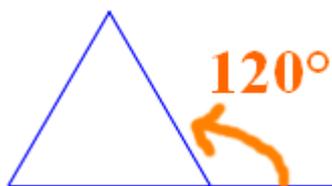
Le premier programme de construction d'un triangle équilatéral ressemble en général à celui-ci ... qui ne fonctionne pas.



Lien : [Quelques constructions géométriques](#)

Dans cette situation simple comme dans d'autres situations plus élaborées (de labyrinthes par exemple), une compréhension de ce qui est en jeu mathématiquement peut passer par un peu d'algorithmique débranchée :

Un triangle équilatéral est tracé au sol et un élève (ou le professeur) avance d'un sommet vers un autre avec le bras droit devant : de combien doit-il tourner pour poursuivre sur le triangle lorsqu'il a atteint le deuxième sommet ?



L'angle de 120° apparaît alors.

L'algorithmique débranchée n'est pas vue comme un préalable mais s'impose ici naturellement pour résoudre un problème. Il convient alors de l'appuyer sur des situations concrètes dans lesquelles l'élève pourra « jouer le rôle du lutin » en intégrant les notions délicates de façon progressive.

Petit à petit, face à de telles situations, l'élève prend naturellement l'habitude de chercher, représenter, modéliser, raisonner pour surmonter les obstacles rencontrés.

2) Les activités rapides

Dans cette optique, s'il est vrai que des activités rapides, des questions flash de début de séances peuvent être un support très pertinent, elles doivent essentiellement entretenir et approfondir des notions déjà installées. Une approche trop abstraite, centrée par exemple sur des calculs mathématiques, ne permettra pas à l'élève de progresser. Par exemple, introduire la boucle par des activités rapides sur la multiplication comme succession d'additions bloquera la compréhension de nombreux élèves. En revanche, lorsque le principe de la boucle est bien installé, ce type de question pourra s'avérer pertinent.

3) Des travaux collaboratifs pour travailler sur des déplacements :

Afin d'amener les élèves à comprendre la nécessité d'instructions claires et précises pour programmer un déplacement, il peut être judicieux de leur faire concevoir un déplacement, les consignes étant alors mises à l'épreuve par d'autres élèves. On est ici dans une forme de travail collaboratif avec des élèves passant du rôle de concepteurs à celui de testeurs.

a) [De la cours à l'ordi](#)

Le travail est d'abord abordé en groupe sous la forme d'algorithmique débranchée : le groupe doit mettre en mots des déplacements, et faire tester ses consignes par un élève

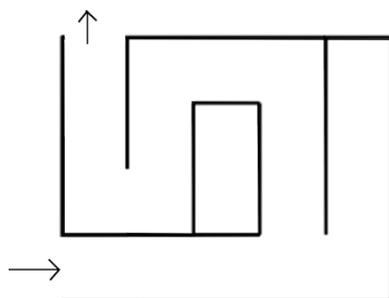
d'un autre groupe. On observe qu'il reste beaucoup d'implicites que le travail sur ordinateur mettra en évidence.

b) [Le Mini GPS](#)

Dans la vie courante, une aide au déplacement est devenue fréquente : le GPS. Sa mise en œuvre, même sous forme simplifiée en classe, montre les éléments mathématiques indispensables alors. Dans l'activité [Mini GPS](#), la programmation en elle-même est simple, mais elle nécessite, au fur et à mesure qu'on cherche à améliorer le programme de base, une belle réflexion sur un usage pertinent des outils mathématiques à disposition.

c) [Labyrinthe](#)

Un travail sur le [labyrinthe](#) a été testé en 5^{ème} à partir d'une collaboration sur un [pad](#).



Le labyrinthe est simple et ne nécessite pas de recherche particulière pour en trouver la sortie.

L'objectif est de se mettre d'accord sur la façon de décrire les actions nécessaires pour arriver à cette sortie.

Ce travail est effectué par groupe de 3 et chaque élève donne une consigne à tour de rôle.

- | | |
|---------------------------|--|
| | Labyrinthe 1 |
| • Avancer de 3,5cm | avancer tout droit jusqu'au premier chemin et tourner a gauche |
| • Tourner de 90° à gauche | puis il faut suivre le chemin jusque a la fin. |
| • Avancer de 3 cm | Tourner a gauche et avanver jusqu'au bout du chemin |
| • Tourner de 90° à gauche | tourner encore un fois a gauche et avancer jusqu'au bout. |
| • Avancer de 2cm | Tourner a droite ? avancer un petit peu et tourner a droite |
| • Tourner de 90° à gauche | avancer jusqu'au bout et vous etes arriver. |
| • Avancer de 2cm | Vous manquez de précision dans les consignes |
| • Tourner de 90° à droite | il faut suivre toujours le même mur jusque au bout |
| • Avancer de 1 cm | aisait le plus dur avec ma technique |
| • Tourner de 90° à droite | tu longes un des mur de départ et tu arivera toujours au bout |
| • Avancer de 3cm | en longant un des mur de d'apart on arives toujours au bout |
| • ~FIN~ | Longuer un mur ? qu'est-ce ? |

Ce travail, commenté si besoin en direct par le professeur (sur fond rouge), aboutit dans chaque groupe à des consignes qui permettent de sortir du labyrinthe. Toutefois, les consignes manquent parfois de précision et nécessiteront une adaptation au logiciel qui ne fonctionne pas en centimètre !

Une telle activité permet de travailler tout particulièrement les compétences raisonner et communiquer.

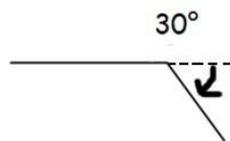
II) Travailler les angles à l'aide de Scratch

3) Travailler les angles avec Scratch

Faire construire des polygones peut aussi être une entrée pertinente pour faire découvrir et travailler les propriétés concernant les angles. [Quelques constructions géométriques](#)

4) Comprendre la gestion des angles sur Scratch

L'activité [angles et parallélisme](#) a mis en évidence le risque d'une entrée trop rapide dans la notion par le côté purement mathématique. Le logiciel Scratch aborde les angles avec l'idée du mouvement. Il est alors inhabituel de visualiser un angle de 30° de la façon suivante :



Il y a là un temps d'appropriation nécessaire pour lequel le recours à l'algorithmique débranchée peut être d'un grand secours. L'objectif affiché de la séance *voir que tourner en degrés ne correspond pas exactement à ce que les élèves connaissent en mathématiques* a donc été difficile à atteindre.

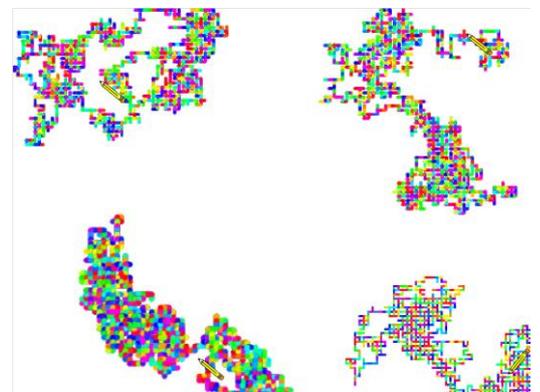
Au-delà des angles en eux-mêmes, se pose aussi la question de l'orientation du lutin. Les déplacements seront en effet réalisés en fonction de l'orientation initiale donnée au lutin : le déplacement horizontal ne va pas de soi ! [Mur et pavages](#)

III) Marches aléatoires

Comme cela est indiqué dans le programme du cycle 4 : *Dès le début et tout au long du cycle 4 sont abordées des questions relatives au hasard, afin d'interroger les représentations initiales des élèves, en partant de situations issues de la vie quotidienne (jeux, achats, structures familiales, informations apportées par les médias, etc.), en suscitant des débats.*

1) [Marches aléatoires](#)

Le travail intervient ici à plusieurs niveaux : d'une part il y a l'aspect programmation en lui-même qui amène à travailler avec l'aléatoire et les déplacements, d'autre part il y a un travail sur la notion de probabilité dans le pari qui peut être fait et l'analyse de la situation ensuite en classe.



2) Articulation travail hors de la classe et dans la classe

La commande **nombre aléatoire entre 1 et 10** permet de simuler des situations de hasard, qu'il convient alors de confronter, en particulier en début de cycle, à une mise en œuvre réelle de type pile ou face ou lancer de dé par exemple. C'est cette démarche qui est utilisée dans [Algo-tests](#) :

a) Un devoir maison introduit du hasard par le biais d'un lancer d'une pièce (pile ou face) et d'un lancer d'un dé, cela initiant des déplacements orientés. Dans ce travail, il n'est pas question de logiciel. Il y a bien une approche algorithmique, il s'agit ici d'algorithmes débranchés.

b) Ces déplacements visualisés sur papier sont à adapter sur Scratch. Il s'agit alors par

exemple de mettre en œuvre de façon pertinente un test , mais aussi de réfléchir sur l'usage des nombres relatifs.

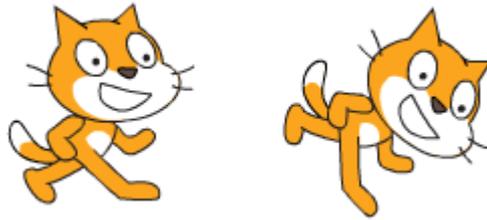
c) Un autre devoir maison demande de comprendre la lecture de scripts simples.



3) [Le juste prix](#)

Deviner un nombre choisi aléatoirement entre 1 et 100. Sur cette page, d'autres pistes sont proposées...

L'INITIALISATION



Dès les premières [constructions géométriques](#) simples se pose le problème de l'initialisation. Il n'est pas immédiat, mais il apparaît dès lors qu'on souhaite renouveler un essai et que le lutin n'est pas revenu à sa position initiale : il a par exemple tourné, ou encore il se trouve au bord de la scène...

Cela est encore plus flagrant lorsque les élèves ont obtenu un algorithme « qui fonctionne » en mode débranché et qu'ils doivent transférer sur Scratch.

[Mini GPS](#) : il faut tenir compte des temps de trajet.

[De la cour à l'ordi](#) : il y a eu des implicites dans le positionnement initial. Sur la cour, les élèves qui ont testé l'algorithme d'un autre groupe se sont orientés convenablement dès le début, personne n'a perçu qu'il fallait préciser la position de départ.

[Labyrinthe](#) : il convient de placer le lutin au bon endroit au départ et de l'orienter convenablement. (Voir aussi [premiers pas](#))

[Mur et pavages](#) : la phase d'initialisation est particulièrement visible dans la partie « pour aller plus loin »

[Tableau de fil](#) : au fur et à mesure de la construction, il faut repositionner le point de départ de construction du segment. Cela peut passer inaperçu en retraçant sur le même trait.

L'initialisation apparaît comme une nécessité et il est alors possible d'en parler avec la classe en indiquant l'importance de cette phase de départ.

DES OBJECTIFS PAS TOUJOURS ATTEINTS

« **L'école est un lieu où on prend des risques sans se mettre en danger** », Philippe Meirieu, journée du numérique de l'académie de Nantes, 23 mars 2016.

S'il est désormais acquis qu'un élève doit bénéficier de stratégies de type essais/erreurs où il apprend autant, voire davantage, d'une erreur bien analysée que des réussites immédiates, il apparaît clairement que les professeurs ayant mis en œuvre différentes activités d'algorithmique en classe ont rencontré des difficultés et sont parfois sortis de la séance avec l'impression désagréable de ne pas avoir atteint l'objectif souhaité !

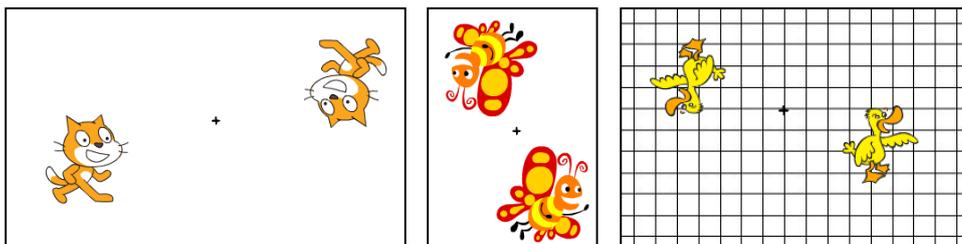
Un élément essentiel est que les élèves n'en tiennent pas rigueur au professeur : ils ont cherché, ont élaboré des conjectures, ont essayé de les valider... Ils ont donc cherché, modélisé, raisonné, communiqué et peut-être représenté et calculé. Bref ils ont fait des

mathématiques ! Le professeur peut donc en toute tranquillité faire un retour sur l'activité, qui sera d'autant plus pertinent qu'il aura pu confronter son analyse avec celle de collègues.

Un exemple : une activité sur la symétrie centrale en 5^{ème}.

Votre mission

Vous devez faire un programme qui permet de faire apparaître un lutin et son symétrique par rapport au point.



Les élèves disposent d'un fichier Scratch avec deux arrière-plans : un blanc et l'autre avec un quadrillage.

Analyse du professeur : *seul un quart des élèves a réussi à programmer quelque chose qui fonctionne (et uniquement ceux qui l'ont fait sans quadrillage). Mais pourquoi n'ont-ils pas réussi ?*

La première raison est d'ordre organisationnel : C'était la première fois que les élèves devaient programmer, sans qu'une réflexion sur le travail à faire n'ait été conduite en classe. Les élèves ont donc choisi avec ou sans quadrillage très rapidement, sans réfléchir à ce que leur choix pouvait entraîner sur la programmation. Ils n'ont pas pris le temps de se poser pour analyser le problème et ce qu'ils avaient à disposition. Aucun élève n'est passé par un travail sur feuille.

La seconde raison est liée au logiciel en lui-même et notamment à la notion de variable : les élèves ont eu de la peine à différencier les variables que l'on affecte nous-mêmes (celles que l'on définit) et celles que l'ordinateur affecte lui-même (« direction », « abscisse x » ...). Le choix des noms des variables a aussi posé problème : il y a eu des confusions entre la variable « x chat » qui avait été définie pour aller positionner le chat et « abscisse x » qui permet d'acquérir l'abscisse du chat.

On peut s'interroger sur la pertinence d'une entrée trop abstraite mettant de plus en œuvre une notion délicate qu'est la variable. C'était sans doute là le principal écueil de cette activité.

LES COMPÉTENCES

Comme cela est indiqué dans les programmes du cycle 4, la mise en œuvre du programme doit permettre de développer les six compétences majeures de l'activité mathématique : chercher, modéliser, représenter, raisonner, calculer, communiquer.

La page Eduscol [le numérique et le socle commun de compétences, de connaissances et de culture](#) précise : « L'usage des outils numériques contribue au renforcement des apprentissages fondamentaux et à la lutte contre le décrochage, facilite la différenciation des démarches et l'individualisation des parcours pour répondre aux besoins de chaque élève. »

Le programme rattache alors ces compétences aux domaines du socle :

Domaine 1 : Les langages pour penser et communiquer

Domaine 2 : Les méthodes et outils pour apprendre

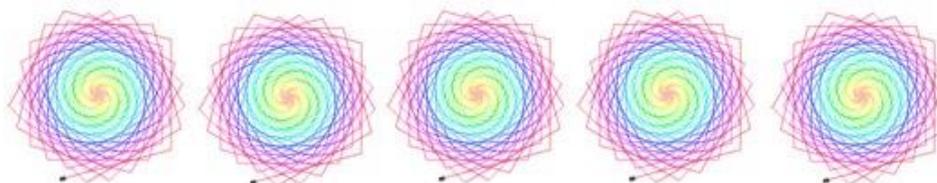
Domaine 3 : La formation de la personne et du citoyen

Domaine 4 : Les systèmes naturels et les systèmes techniques

Domaine 5 : Les représentations du monde et l'activité humaine

Chercher : domaines 2 et 4	Domaine 1 : modéliser, représenter, communiquer
Modéliser : domaines 1, 2 et 4	Domaine 2 : chercher, modéliser, raisonner
Représenter : domaines 1, 4 et 5	Domaine 3 : raisonner, communiquer
Raisonner : domaines 2, 3 et 4	Domaine 4 : chercher, modéliser, raisonner, calculer
Calculer : domaines 1 et 4	Domaine 5 : représenter
Communiquer : domaines 1, 3 et 4	

On peut synthétiser ces croisements compétences/domaines comme cela est proposé dans la page suivante.



	Domaine 1 Les langages pour penser et communiquer	Domaine 2 Les méthodes et outils pour apprendre	Domaine 3 La formation de la personne et du citoyen	Domaine 4 Les systèmes naturels et les systèmes techniques	Domaine 5 Les représentations du monde et de l'activité humaine
Chercher		<ul style="list-style-type: none"> - S'engager dans une démarche - Tester, essayer 		<ul style="list-style-type: none"> - Extraire l'information, reformuler, organiser, ... - S'engager dans une démarche - Tester, essayer - Décomposer en sous-problèmes 	
Modéliser	<ul style="list-style-type: none"> - Passer d'un langage courant à un langage mathématique puis/ou à un langage informatique 			<ul style="list-style-type: none"> - Modéliser - Mettre en œuvre la proportionnalité - Décrire des phénomènes - Lier cause(s) et effet(s) - Concevoir un déplacement 	
Représenter	<ul style="list-style-type: none"> - Compléter et lire un tableau. - Choisir et mettre en relation des cadres (numérique, algébrique, géométrique, algorithmique, ...) 			<ul style="list-style-type: none"> - Interpréter des résultats statistiques et les représenter graphiquement 	<ul style="list-style-type: none"> - Côtayer le hasard - Imaginer, concevoir et réaliser des productions de natures diverses
Raisonner		<ul style="list-style-type: none"> - Justifier ses affirmations - Rechercher la validité des informations - Résoudre des problèmes nécessitant l'organisation de données multiples - Progresser collectivement 	<ul style="list-style-type: none"> - Progresser collectivement - Justifier ses affirmations - Rechercher la validité des affirmations 	<ul style="list-style-type: none"> - Passer de la perception au contrôle par les instruments, y compris numériques - Mener une démarche d'investigation 	
Calculer	<ul style="list-style-type: none"> - Utiliser le langage algébrique 			<ul style="list-style-type: none"> - Calculer en utilisant un langage algébrique - Mettre en œuvre des calculs avec un logiciel - Contrôler la vraisemblance d'un résultat obtenu 	
Communiquer	<ul style="list-style-type: none"> - Faire le lien entre le langage naturel et le langage algébrique ou algorithmique - Distinguer les spécificités des différents langages - Expliquer à l'écrit ou à l'oral une démarche, un raisonnement, un calcul 		<ul style="list-style-type: none"> - Distinguer ce qui est objectif de ce qui est subjectif - Lire, interpréter, produire 	<ul style="list-style-type: none"> - Rendre compte de sa démarche en utilisant le langage scientifique à bon escient 	

DES PERSPECTIVES POUR L'AN PROCHAIN

Plusieurs autres thèmes ont été abordés dans le groupe comme autant de pistes ouvertes que nous souhaitons approfondir l'an prochain.

1) Les moments de régulations

Lors d'une activité, un groupe plus ou moins important d'élèves peut rencontrer des blocages. Comment anticiper ces blocages ? Comment intervenir individuellement ? Collectivement ?

De façon générale donc, quelles sont les pistes pour gérer ces moments de régulation tant sur le plan de l'organisation que des contenus ?

2) Communication sur un programme

Cet aspect peut d'ailleurs être en lien avec le précédent : lors de la réalisation d'un script simple, une erreur est assez facilement détectable. Dans le cas d'un programme plus élaboré, l'élève doit être à même de présenter son travail : quelle(s) stratégie(s) sont mises en œuvre ? Quel est le rôle de telle ou telle variable ? Comment les différents éléments du script s'organisent-ils ? ...

Quelle place peut-on donner au travail partagé : décomposition d'un programme en sous-programmes réalisés par différents élèves et réunis ensuite ?

3) Trace écrite

Un moment important sera celui de la trace écrite. En quoi peut-elle consister ? Peut-elle se réduire à des exemples significatifs ? À quel moment cette trace écrite peut-elle apparaître ?

4) EPI, AP

Les heures utilisées en AP ou en EPI peuvent-elles être valorisées par des entrées algorithmiques ? Dans quelle mesure l'algorithmique peut-elle être un support à l'interdisciplinarité ? Quelle peut-être la place d'une pédagogie de projet avec l'apport de l'algorithmique ?

5) L'évaluation

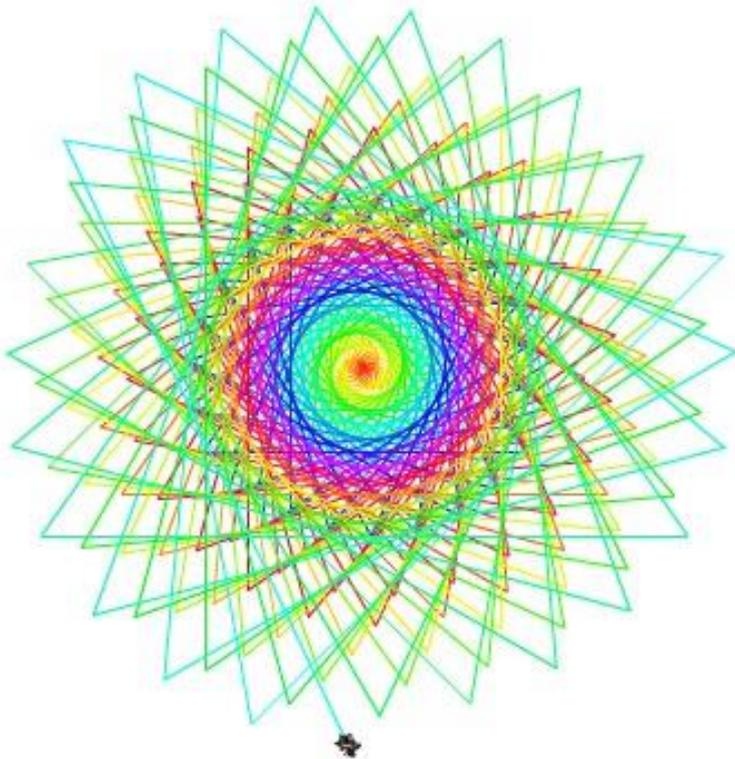
Comment mettre en œuvre une évaluation de l'algorithmique ? Quelle est la place des compétences dans les activités que l'on peut proposer ?

6) Le lycée

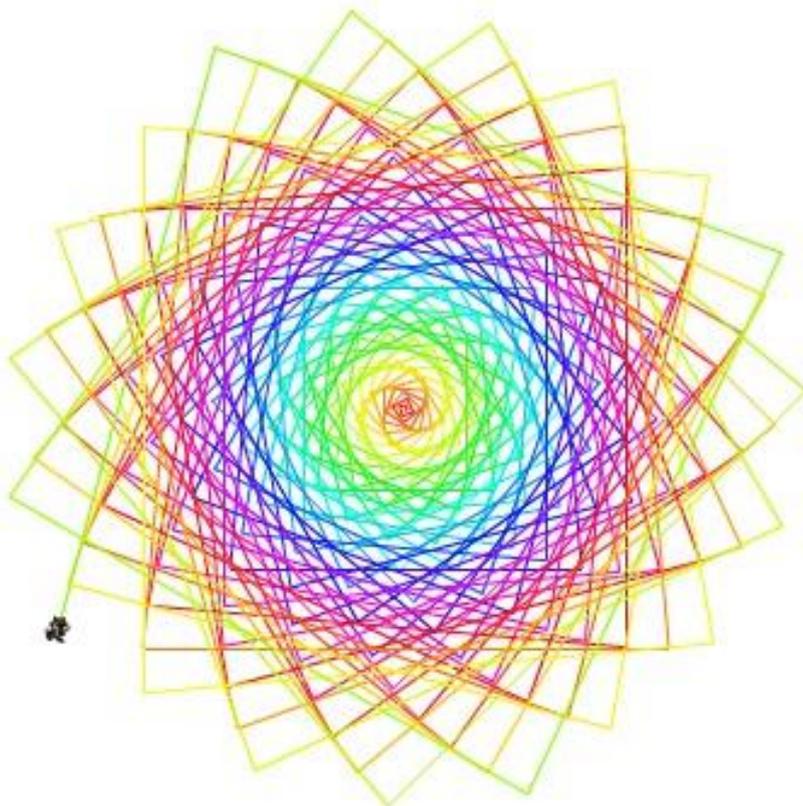
Quelles passerelles entre l'enseignement de l'algorithmique en collège et celui du lycée ? Quels éléments de l'enseignement d'exploration ICN en 2^{nde} sont initiés dans le cadre des programmes de collège ?

ANNEXE 1

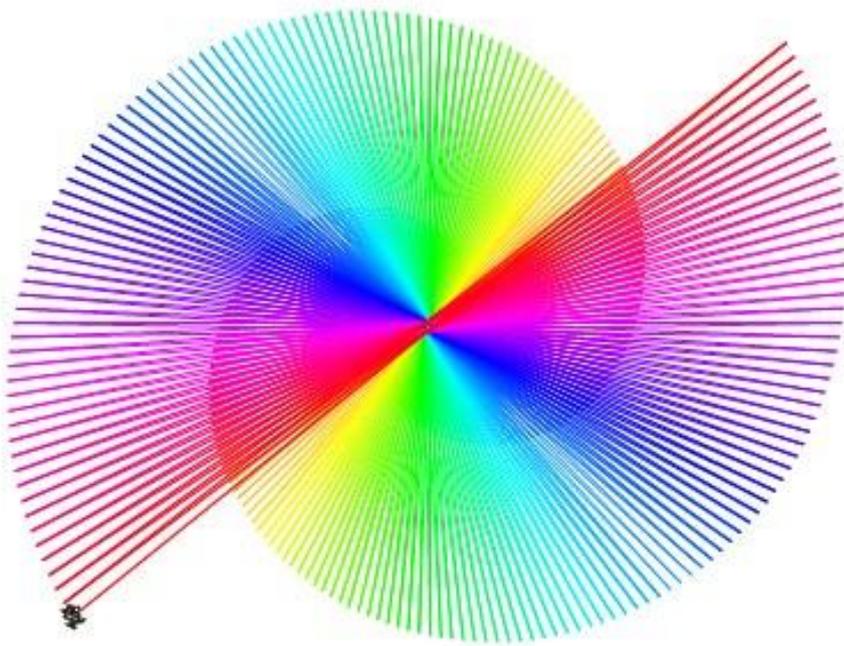
Programmes ayant servi à la construction des images lors de changements de paragraphe. On observe qu'il s'agit pratiquement du même programme avec juste quelques variantes. La recherche de structures communes pouvant resservir dans différents programmes est un objectif intéressant à poursuivre !



```
quand cliqué
aller à x: 0 y: 0
mettre la couleur du stylo à 0
effacer tout
stylo en position d'écriture
s'orienter à 90
mettre itération à 1
répéter 300 fois
  avancer de itération
  tourner de 125 degrés
  ajouter à itération 1
  ajouter 1 à couleur du stylo
relever le stylo
```

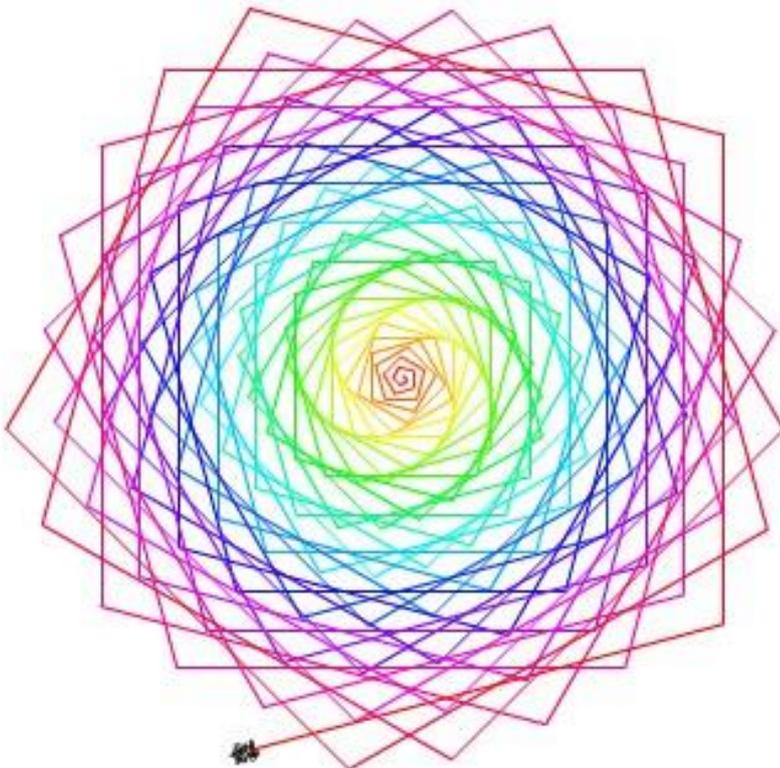


```
quand cliqué
aller à x: 0 y: 0
mettre la couleur du stylo à 0
effacer tout
stylo en position d'écriture
s'orienter à 90
mettre itération à 1
répéter 250 fois
  avancer de itération
  tourner de 95 degrés
  ajouter à itération 1
  ajouter 1 à couleur du stylo
relever le stylo
```



```

quand cliqué
  aller à x: 0 y: 0
  mettre la couleur du stylo à 0
  effacer tout
  stylo en position d'écriture
  s'orienter à 90
  mettre itération à 1
  répéter 400 fois
    avancer de itération
    tourner de 181 degrés
    ajouter à itération 1
    ajouter 1 à couleur du stylo
  relever le stylo
  
```



```

quand cliqué
  aller à x: 0 y: 0
  mettre la couleur du stylo à 0
  effacer tout
  stylo en position d'écriture
  s'orienter à 90
  mettre itération à 1
  répéter 200 fois
    avancer de itération
    tourner de -75 degrés
    ajouter à itération 1
    ajouter 1 à couleur du stylo
  relever le stylo
  
```

ANNEXE 2

Ce tableau reprend les activités mises en ligne en précisant les domaines et compétences mises en œuvre qui dominent dans ces activités ainsi que des pistes de différenciations associées.

Domaine 1 : Les langages pour penser et communiquer (plus précisément : comprendre, s'exprimer en utilisant les langages mathématiques et informatiques).

Domaine 2 : Les méthodes et outils pour apprendre (plus précisément : coopérer et réaliser des projets).

Domaine 3 : La formation de la personne et du citoyen (plus précisément : prendre des initiatives).

Domaine 4 : Les systèmes naturels et les systèmes techniques (plus précisément : pratiquer une démarche scientifique ; concevoir, créer et réaliser).

Domaine 5 : Les représentations du monde et l'activité humaine (plus précisément : pratiquer une démarche scientifique ; concevoir, créer et réaliser).

Activités	Compétences	Différenciation
Quelques algorithmes simples	Modéliser Représenter	Les entrées progressives permettent très vite des programmes assez élaborés dès lors qu'on entre dans une forme d'optimisation.
De la cour à l'ordi	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer	La liberté donnée à la façon de programmer autorise de multiples possibilités générant des difficultés variables.
Labyrinthe Premiers pas	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer	Le travail collaboratif permet à chaque trio d'élèves d'avancer à son rythme dans la partie algorithme débranchée, puis chacun met en œuvre à sa façon sur le logiciel. Les labyrinthes plus élaborés sont là pour les plus rapides. On peut aussi imaginer une automatisation pour sortir du labyrinthe.
Mini-GPS	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer ; Calculer	Le travail de groupe, en particulier lors de la validation de conjectures. La mise en œuvre « sur le terrain » lors d'une sortie pédagogique.
Quelques constructions géométriques	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer ; Calculer	Travail en binôme. Complexification possible des figures à réaliser et/ou mise en œuvre de nouvelles fonctions de programmation (boucle, test, ...)
Algo-tests	Chercher ; Modéliser ;	Passage par le mode débranché

	Représenter ; Raisonner ; Communiquer ; Calculer	Progressivité, en autonomie. Observation et conjecture.
Marches aléatoires		
Commande au restaurant	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer ; Calculer	Choix du logiciel le plus adapté : pas nécessairement le logiciel Scratch.
Quadrilatères	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer	Possibilité de restreindre à la reconnaissance d'un seul quadrilatère avant d'élargir : quelles questions poser alors pour savoir si c'est un carré, par exemple ? Passage par un algorithme débranché (papier/crayon) avant de programmer. Travail en groupes.
Angles et parallélisme	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer ; Calculer	Modification de la situation de départ. Travail en groupes.
Mur et pavages	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer	Travail évolutif sur plusieurs séances. Structures élaborées s'appuyant sur des éléments simples.
Tableau de fil	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer ; Calculer	Travail sur l'erreur Structures élaborées s'appuyant sur des éléments simples Évolution prévues pour les plus rapides
Somme d'entiers consécutifs	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer ; Calculer	Laisser vivre plusieurs approches (utilisation ou non de variables, nombre de variables, ...) Travail sur l'erreur
La fourmi de Langton	Chercher ; Modéliser ; Représenter ; Raisonner ; Communiquer ; Calculer	Temps de travail en algorithmique débranchée Travail collaboratif

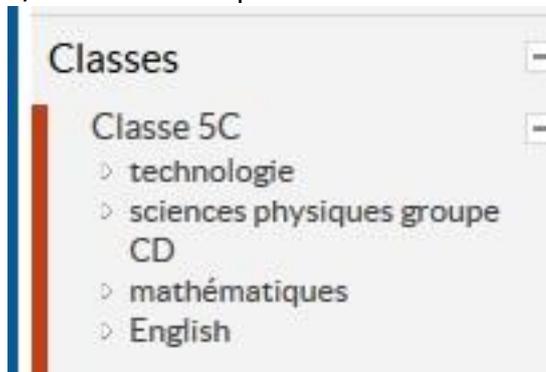
ANNEXE 3

Un pad est un espace de travail collaboratif.

Sur l'espace e-lyco de l'académie de Nantes, on peut choisir cette option.

On peut insérer un pad par exemple dans la rubrique classe.

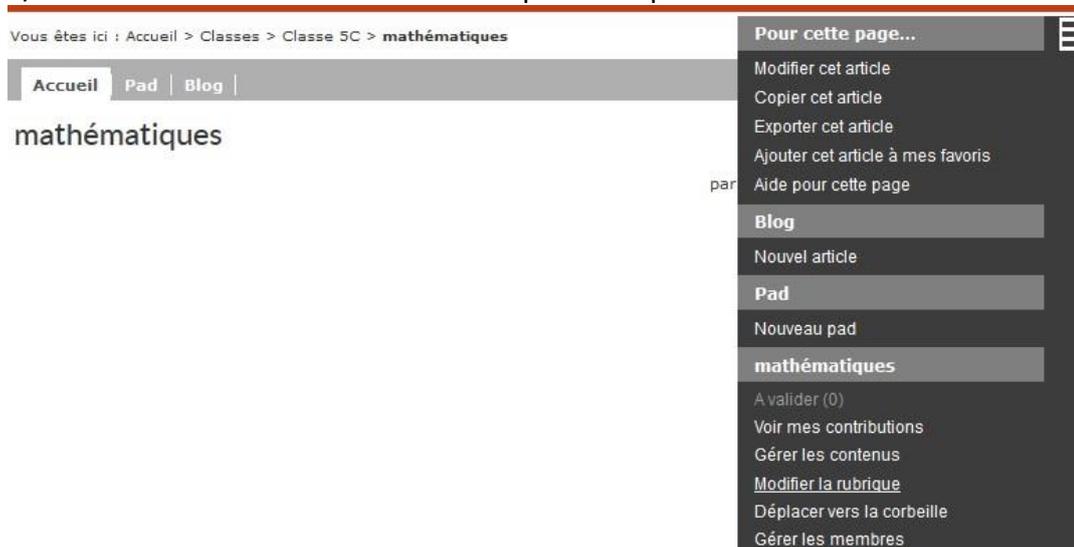
1/ Ouvrir la rubrique classe



Et créer si besoin une sous-rubrique disciplinaire.



2/ Se rendre ensuite dans la sous-rubrique créée pour la modifier.



Et cocher le « Service pad » dans l'onglet Services

Services

- Service Pad
- Service Blog
 - Activer le flux RSS
- Service Agenda
- Service Forum
- Service Chat
- Service Dossiers

Ajouter une catégorie d'article

Catégorie (*)

Aucune catégorie définie

3/ Lorsque tout est validé, se rendre dans l'onglet Pad et créer un nouveau pad

Accueil Pad Blog

9 pads | Trier par **Ordre alphabétique**

4/ On peut créer autant de pad que nécessaire en les nommant par exemple avec les noms des élèves formant les groupes.

9 pads | Trier par **Ordre alphabétique**

Adil ; Esther ; Solenn
Créé par YANNICK DANARD | Dernière contribution le 03 déc. 2015 à 16:26

Camélia ; Titouan ; Jeanne
Créé par YANNICK DANARD | Dernière contribution le 03 déc. 2015 à 16:35

Flavien ; Melvin ; Anaïs
Créé par YANNICK DANARD | Dernière contribution le 03 déc. 2015 à 16:32

Laura ; Domnin ; Louise
Créé par YANNICK DANARD | Dernière contribution le 03 déc. 2015 à 17:32

Maëlys ; Jean-Baptiste ; Gabin
Créé par YANNICK DANARD | Dernière contribution le 03 déc. 2015 à 16:37

Malcolm ; Quentin ; Éloïse
Créé par YANNICK DANARD | Dernière contribution le 03 déc. 2015 à 16:38

Annexe 4 :

Quel logiciel pour quels usages ?

L'enseignement des mathématiques s'appuie désormais sur 3 logiciels principaux : tableur, géométrie dynamique, algorithmique. Beaucoup d'activités que l'on pouvait réaliser sur tableur ou sur un logiciel de géométrie dynamique peuvent désormais être effectuées sur le logiciel d'algorithmique. On peut toutefois s'interroger sur la pertinence des usages.

Le tableau ci-dessous, qui ne se veut en aucun cas exhaustif, propose une première réflexion sur le logiciel (ou les logiciels) le plus adapté pour aborder telle ou telle notion.

Thèmes du programme	Tableur	Géométrie dynamique	Algorithmique
Figures simples : polygones		La construction de triangles ou quadrilatères par exemple seront abordés de façon complémentaire avec ces deux logiciels : <ul style="list-style-type: none"> - Le logiciel de géométrie dynamique permettra par exemple de tester la « robustesse » d'une figure : garde-t-elle ses propriétés lorsqu'on bouge un point ? - Le logiciel d'algorithmique invite à repenser la figure en termes de déplacements, et apporte un regard différent, par exemple sur les angles. 	
Figures simples : cercle, arc de cercle		Le logiciel de géométrie dynamique est plus adapté.	Ce logiciel permet la construction de forme qu'on peut associer au cercle, mais n'est pas idéal pour construire une réflexion mathématique sur ce type de figures.
Variables, lettres	Une cellule nommée et appelée peut être une approche très pertinente de la notion de variable.	L'usage du curseur en particulier permet une belle approche de la notion de variable.	La variable informatique arrive en complément de la variable mathématique : elle n'a cependant pas la même signification et il convient de travailler aussi sur les différences de sens.
Calculs numériques	Le tableur est très adapté au calcul numérique : <ul style="list-style-type: none"> - Appeler une cellule nommée - Etirer - Utiliser les fonctions spécifiques : SOMME, MIN, MAX, ... 		On peut effectuer des calculs numériques mais utiliser le logiciel d'algorithmique à ce seul usage (du type : construire un exerciceur sur la tables de multiplication) peut être assez pauvre du point de vue des compétences développées : en particulier, ce type de travaux ouvre peu la voix à la différenciation lors de la construction du programme.
Algorithmique	L'usage de test conditionnel dans des formules est une entrée algorithmique.		Ce logiciel s'impose : boucles, tests conditionnels, variables, blocs...pour la géométrie et les jeux en particulier.
Aléatoire	Logiciel adapté avec les fonctions ALEA et ALEA.ENTRE.BORNES	Logiciel adapté avec la fonction RANDOM()	Logiciel adapté avec le bloc nombre aléatoire.

Certains logiciels permettent en outre l'utilisation du calcul formel et de géométrie dans l'espace.