



# TraAM 2017-18 : L'algorithmique avec des tracés

publié le 24/05/2018 - mis à jour le 06/06/2018

---

## Descriptif :

Faire une dernière activité de cycle 4 avec un contenu centré sur les notions d'algorithmique.

---

## Sommaire :

- Caractéristiques du scénario
  - Déroulement du scénario
  - Déroulements pas à pas et impressions
  - Retour d'expérience :
  - Évaluation :
- 

### ● Caractéristiques du scénario

Ce scénario est pensé comme étant ce qu'il serait souhaitable que les élèves sachent faire à la fin du cycle 4 concernant la programmation et l'algorithmique et ainsi permettre une bonne entrée au lycée.

#### ○ Thématique

Les notions qui normalement devraient être acquise en algorithmique et en programmation à la fin du cycle 4.

#### ○ Niveau concerné

Fin de cycle 4.

#### ○ Compétences mobilisées

Chercher, modéliser, Reasonner, Communiquer.

#### ○ Problématique

Comment utiliser les ressources de bases de l'algorithmique : boucles, variables et fonctions informatiques, instructions conditionnelles, pour produire des scripts concis permettant de faire des tracés ? Comment faire pour que ces scripts aient en plus une bonne adaptabilité à des modifications de spécifications des tracés ?

#### ○ Nombre d'heures envisagées

3-4 heures avec le logiciel scratch si possible en travail individuel. Puis une évaluation.

#### ○ Outils et ressources

Des équipements sur lesquels on peut utiliser scratch pour pouvoir travailler sur un unique fichier.

**En plus si possible :** Un outil pour afficher les productions des élèves au vidéo-projecteur (un logiciel de prise en main des ordinateurs à distance comme eopotes par exemple) afin d'engendrer des débats autour des idées ou des erreurs.

### ● Déroulement du scénario

Une suite de tracés est proposée. Il faut les reproduire à l'aide de scripts dans scratch. Les difficultés vont crescendo : besoin de boucles, puis de fonctions informatiques sans variable peut-être au début, puis avec

variable(s) par la suite, variables informatiques, puis instructions conditionnelles.

Les élèves ont un tracé géométrique à reproduire en arrière plan. Ils doivent repasser par dessus. Ils ont donc un retour instantané sur la justesse de leurs scripts.

Les élèves sont dans une salle où ils peuvent utiliser le logiciel scratch de façon individuelle si possible.

Ils ont à disposition un fichier scratch préparé, à compléter :

(appuyez sur les touches numériques de votre clavier après avoir cliqué sur le drapeau vert)



1. L'appui sur les touches numériques du clavier changent l'arrière plan, l'élève voit ainsi l'objectif à atteindre.
2. Le début du script est fait pour le premier tracé..
3. Le lutin est un crayon avec une flèche pour indiquer son orientation.

L'appuie sur la touche correspondant à l'arrière plan doit déclencher le tracé du dessin.

Quand l'objectif est atteint, l'élève prévient l'enseignant et passe au dessin suivant.

L'enseignant peut alors commenter avec lui sa solution :

1. Est-ce que toutes les parties du scripts sont utiles ?
2. Est-ce qu'il y a moyen de diminuer le nombre de blocs ?
3. Est-ce que le script peut s'adapter si le dessin est légèrement différent ?
4. Quels ont été les obstacles ?

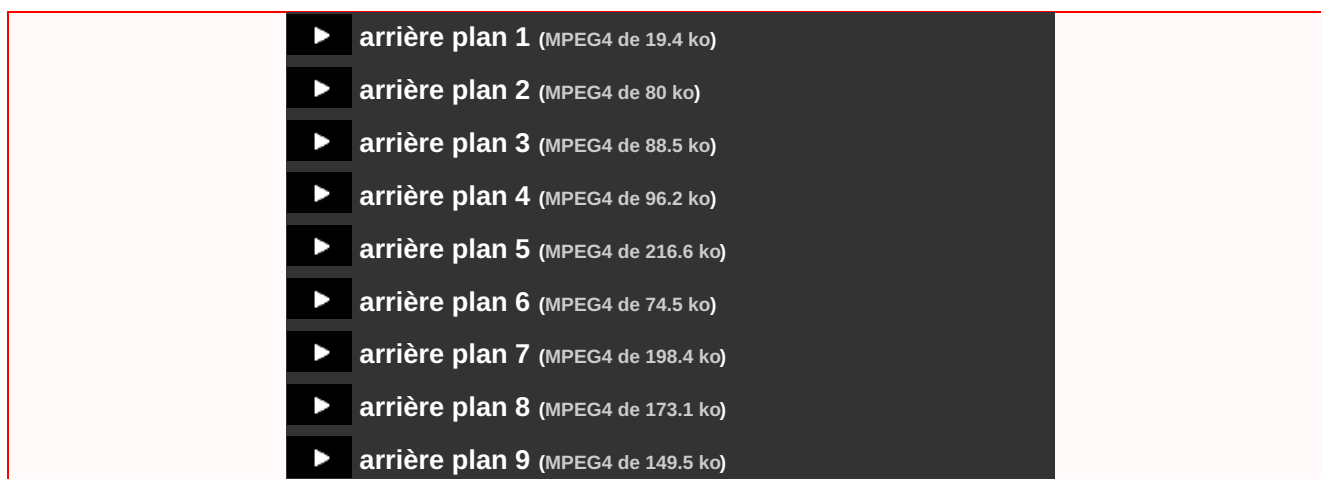
Progression de la difficulté (si on ne tient pas compte des blocs d'initialisation) :

1. Les blocs bleus peuvent suffire.
2. Le bloc "répéter" raccourcit grandement le script.
3. Une fonction scratch, la création d'un bloc violet, peut faciliter le script.
4. Une fonction scratch, la création d'un bloc violet, avec une variable est plus pratique.
5. L'utilisation d'instructions conditionnelles est normalement nécessaire.
6. L'utilisation d'une variable répond efficacement au problème.
7. Une fonction scratch avec plusieurs variables simplifie le problème.

Les dessins sont composés de carrés. C'est simple à tracer et donc l'attention est reportée sur le reste, c'est à dire les notions d'algorithmique.

### Les aides possibles :

Avoir le dessin "fixe", peut bloquer certains élèves : "Par où je vais commencer ?". On peut donc mettre à disposition les captures vidéos de tracés.



Toutefois, un inconvénient des vidéos, c'est qu'elles orientent la façon de penser le tracé et freinent ainsi l'originalité des scripts. Elles sont donc à utiliser en dernier recours.

On peut rappeler les techniques de détection des erreurs :

- ▶ Enlever tous les blocs puis les rajouter un par un en exécutant le script à chaque fois éventuellement en mode turbo. (accessible à partir du menu "Édition")
- ▶ Isoler des blocs pour les tester séparément.

On peut aussi rappeler les réflexes à avoir :

- ▶ Si les mêmes blocs s'enchaînent plusieurs fois dans le même ordre alors il est temps de songer à placer un bloc répéter ou à créer une fonction scratch supplémentaire (un bloc violet).
- ▶ Si on réutilise la même suite de blocs en remplaçant une valeur par une autre, c'est sans doute l'occasion de créer une fonction scratch (un bloc violet) avec une variable.
- ▶ Verbaliser ce que l'on veut obtenir.

Les erreurs classiques :

- ▶ Oublier d'initialiser l'orientation. Ce qui peut donner lieu à des dessins qui partent dans tous les sens. Cela arrive quand les élèves arrêtent un script avant la fin puis relancent le script.
- ▶ Faire le mauvais choix entre les déplacements relatifs ou absolus et même chose pour l'orientation. Ce cas de figure n'est pas évident à gérer avec les élèves car ils peuvent avoir passé beaucoup de temps à réaliser le tracé et lorsqu'ils se rendent compte que leur script n'est pas performant, il peut y avoir un gros découragement. Certains élèves peuvent affronter la prise de conscience des faiblesses de leurs scripts avec courage, mais les plus fragiles doivent être détectés avant d'être trop avancés dans leurs scripts dans la mesure du possible. Après, s'ils veulent continuer sur leur idée, au moins ils auront été prévenus...

### ● Déroulements pas à pas et impressions

La mise en œuvre est plutôt assez simple : le fichier scratch est mis à la disposition des élèves. La consigne est la suivante : "Les touches du clavier de 1 à 9 correspondent à un arrière plan différent sur lequel il y a le tracé d'une figure. L'objectif est de repasser avec le lutin en forme de crayon sur chacun des tracés. Quand on appuie sur la touche du clavier, il faut que le script lance le tracé automatiquement et uniquement celui qui correspond à l'arrière plan. Pour faire simple : il faut repasser en noir ce qui est en bleu."

Les élèves sont en autonomie.

Les interventions de l'enseignant ne sont collectives que si le point à traiter concerne une grande majorité des élèves.

Les élèves doivent se débrouiller pour trouver les coordonnées de départ. On peut rappeler aux plus fragiles où sont indiquées les coordonnées actuelles du lutin sélectionné.

Il leur sera peut-être utile de créer un script qui efface les tracés. À eux de prendre l'initiative.

**Voici les différents tracés à effectuer :**

1.



Le carré peut très bien se faire avec uniquement des blocs bleus. (les élèves qui font comme cela sont à repérer) Évidemment, ce qui est attendu c'est au moins un bloc "répéter". Lors des expérimentations, personne n'a créé de fonction (bloc violet) du type "tracer un carré" à ce stade.

Pour ceux qui n'ont utilisé que des blocs bleus, on peut toujours leur demander combien ils ont de blocs à changer si on veut un carré de 50.

*Pour les rapides : on pourra revenir sur cet arrière plan ultérieurement et demander un carré sur la pointe par exemple, une “fenêtre” de quatre carrés, un carré avec ses diagonales de tracées.*

2.



Quelques rares élèves n'ont encore utilisé que des blocs bleus. C'est vrai qu'il est très facile de faire des duplications. On peut alors leur dire que s'ils dupliquent, c'est qu'il y a une répétition...

Quelques rares élèves n'ont encore utilisé que des blocs bleus. C'est vrai qu'il est très facile de faire des duplications. On peut alors leur dire que s'ils dupliquent, c'est qu'il y a une répétition...

On peut là encore poser des questions du type : Et si on veut plutôt 10 carrés de 20 espacés de 5, combien faut-il modifier de blocs ?

*Pour les rapides : on pourra revenir sur cet arrière plan ultérieurement et demander une ligne avec un nombre de carrés différents, avec une taille différente et un espacement différent.*

3.



Le besoin de créer une fonction avec pour variable la taille du carré ne se fait pas encore ressentir. Certains ne connaissent pas forcément cette possibilité qu'il faudra peut-être rappeler après le tracé suivant. Par contre, d'autres créent une fonction “tracer un carré” et “tracer un petit carré” ou “tracer un carré de 40”...

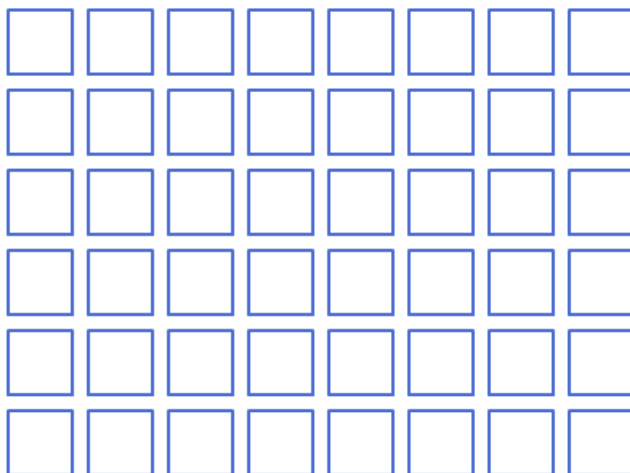
4.



Pour ce tracé, les laisser écrire un très long script pour arriver à se demander s'il n'y a pas plus court est très formateur. Il a fallu rappeler à certains comment créer et utiliser une variable en précisant que leur nom doit être évocateur : surtout pas “x” mais plutôt “côté du carré”.

*Pour les rapides : on pourra demander une ligne ayant un axe de symétrie verticale, puis un axe horizontale, puis les deux.*

5.



On peut indiquer que si une fonction du type “tracer une ligne de 8 carrés” est définie, le script principal est

beaucoup plus lisible. On peut même suggérer une petite fonction “retour à la ligne”. Tous leurs scripts créés précédemment peuvent devenir des fonctions scratch avec ou sans variables. Il a fallu montrer à certains élèves maniant un peu trop vite la duplication, qu’une boucle pouvait être utilisée. On peut leurs déplacer avec la souris, à côté de leur long script, le groupe de blocs qui se répètent.

*Pour les rapides : on pourra demander le même type de modifications que pour le tracé 2.*

6.



Pour ce tracé, les scripts peuvent être assez différents suivant l’endroit où le bloc “stylo en position d’écriture” est placé ou si l’élève a choisi la couleur blanche pour le carré que l’on ne voit pas, ou s’il saute directement le carré... Il a fallu “imposer”, “proposer” comme un challenge à certains élèves l’utilisation d’instructions conditionnelles après leur avoir demandé comment il ferait pour que ce soit un autre carré qui ne soit pas tracé et avec un nombre de carrés différent ?

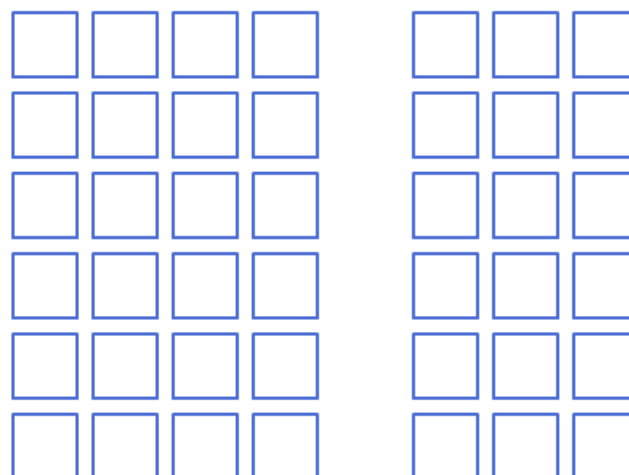
Remarque :

Un élève a très bien réussi le tracé sans instruction conditionnelle. Et son script s’adaptait à un changement de la position de l’espace vide et du nombre de carrés. Pour remplir un des objectifs de l’activité “vérifier la capacité à utiliser des instructions conditionnelles” sans le forcer à changer son script qui fonctionnait bien, il lui a été demandé de tracer des carrés “jusqu’à” ce qu’il n’y ait plus assez de place pour en tracer un entier . Ce qu’il a très bien réalisé en utilisant l’abscisse du lutin, la taille du carré et “répéter jusqu’à”. (l’utilisation de “jusqu’à” dans la phrase permet d’orienter la création du script car sinon, on peut aussi trouver par un calcul le nombre de carrés entiers maximal que l’on peut tracer)

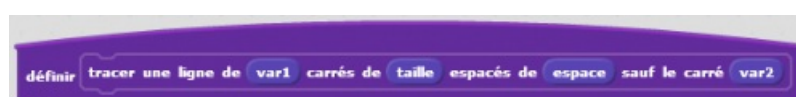
On peut aider pour ce tracé avec des questions du type : Qu’as tu fait pour connaître la position de l’emplacement vide ? (réponse attendue : J’ai compté.) Comment faire compter l’ordinateur ? On peut aussi lui demander combien il y a de carrés tracés.

Oraliser, verbaliser, a été nécessaire pour les élèves les plus fragiles.

7.



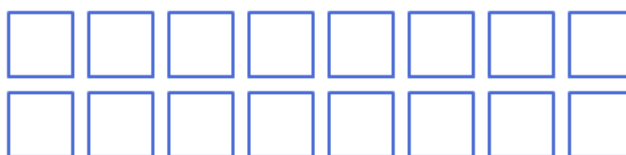
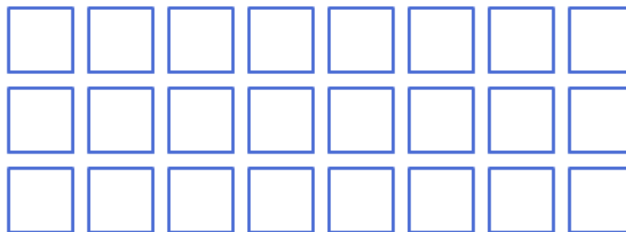
Les tracés 7, 8 et 9 sont plus compliqués mais n’utilisent pas forcément de notions algorithmiques supplémentaires. Le 9 sans instructions conditionnelles va être très dur à tracer. Ils sont à réserver aux élèves les plus rapides en leur suggérant la création de fonctions scratch avec jusqu’à 4 variables comme par exemple : “tracer une ligne de var1 carrés de taille espaces de espace sauf le carré var2”



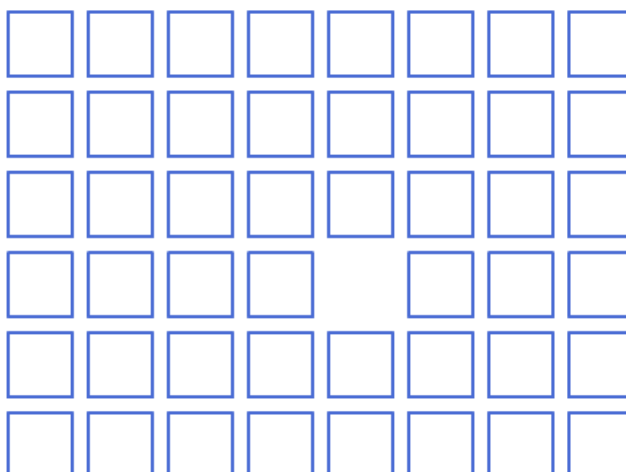
Avec de telles fonctions, les scripts deviennent beaucoup plus lisibles.

Remarque : “carré” serait un très mauvais nom pour une fonction, utiliser plutôt “tracer un carré” car on **commande** une machine. Cela nous rapproche du lycée avec les **langages textuels** qui utilisent une **console** dans laquelle on écrit des lignes de **commandes**.

8.



9.



On peut aussi prolonger le 9 en demandant une grille avec une colonne et une ligne vide ou, avec l'intersection de la ligne et de la colonne vide qui a un carré.

Avec toujours cette idée de savoir si les scripts vont s'adapter à des valeurs différentes concernant les colonnes et les lignes vides, leur nombre, si on remplace le carré par autre chose...

On pourrait aussi envisager de jouer sur les couleurs, de mettre les carrés dont le numéro est un multiple d'un nombre d'une certaine couleur, de refaire le crible d'Ératosthène...

On pourrait aussi jouer sur la taille des carrés : essayer de paver un rectangle (plus petit que l'espace de travail de scratch pour ne pas avoir de problèmes avec les tracés trop près des bords) avec des carrés les plus grands possibles ou avec le plus de carrés possible...

#### ● Retour d'expérience :

Peu de problèmes de motivation car les tracés sont d'apparence si simple que les élèves s'accrochent à les reproduire. Il faut quand même surveiller ceux qui restent vraiment trop longtemps sur un même tracé.

Bien montrer les différentes productions à la fin de l'activité, avec leurs différences et leur adaptabilité. On peut faire expliquer leurs choix à certains.

Appuyer sur le fait que si une fonction scratch n'existe pas, il faut la créer : définir un nouveau bloc violet. Le nom de la fonction doit commencer si possible par un verbe, le script en devient plus clair.

Tous ces scripts pour un même lutin, peuvent rendre la lecture difficile si l'élève n'est pas un minimum organisé. Pour "aérer" les scripts, on peut donc dupliquer le lutin crayon mais attention aux mélanges. Après la duplication, il faut modifier les événements si besoin. Changer la couleur du crayon ou ajouter un numéro au costume du lutin crayon permet d'éviter les confusions et permettre de savoir ainsi quel lutin fait quoi.

Si les élèves réussissent cette activité, ils pourront commencer l'année de seconde avec le même genre d'activité mais dans un environnement python :

Dans cette activité, scratch est censé être connu, la difficulté est algorithmique.

En seconde, les bases de l'algorithmique sont censées être connues, la difficulté est la découverte d'un environnement Python, sa déclaration de fonctions informatiques...

lien à venir...

### ● Évaluation :

On peut proposer un fichier sur le même principe pour une évaluation.



Pour l'évaluation, on peut utiliser les critères suivants :

- ▶ L'objectif n'est pas atteint si même le script de la touche un n'est pas correctement écrit ;
- ▶ Il est partiellement atteint si les scripts des trois premières touches ne sont pas complètement correctes ;
- ▶ Il est atteint si les scripts des trois premières touches sont correctes ;
- ▶ Il est dépassé si les scripts de la touche quatre sont correctes.