



TraAM 2017-18 : Géométrie repérée et algorithmique

publié le 24/05/2018

Sommaire :

- Caractéristiques du scénario
- Déroulement du scénario
- Prolongements et évaluation :
- Documents de référence :

● Caractéristiques du scénario

○ Thématique

Construire la notion de fonction en programmation Python.

○ Niveau concerné

Pour une classe de seconde. Le calcul des coordonnées du milieu d'un segment et de la distance de deux points connaissant leurs coordonnées ne permet pas de l'intégrer dans le cycle 4.

○ Compétences mobilisées

Raisonner (élaborer une démonstration), Représenter, Modéliser.

○ Problématique

Comment utiliser les fonctions Python pour construire un programme capable de déterminer si un quadrilatère est un parallélogramme, un rectangle ?

L'objectif est de découvrir l'outil fonction de Python. Cette découverte s'appuie sur les notions de géométrie du cycle 4 et les contenus de géométrie repérée de seconde.

○ Nombre d'heures envisagées

2 heures en salle informatique ou accompagnées d'un poste par groupe de 4 élèves.

○ Outils et ressources

On utilise un questionnaire Googleforms pour tester et former les élèves en amont.

● Déroulement du scénario

Avant d'aborder ce parcours, il convient d'avoir bien installé chez les élèves la notion de variable en algorithmique mais aussi les contenus des programmes autour de la géométrie repérée. Les élèves doivent déjà avoir eu une initiation éclairée du langage Python.

Ces petites missions peuvent permettre d'aborder le langage en autonomie ou en AP :

 [Listes de commandes Python](#) (PDF de 205.8 ko)

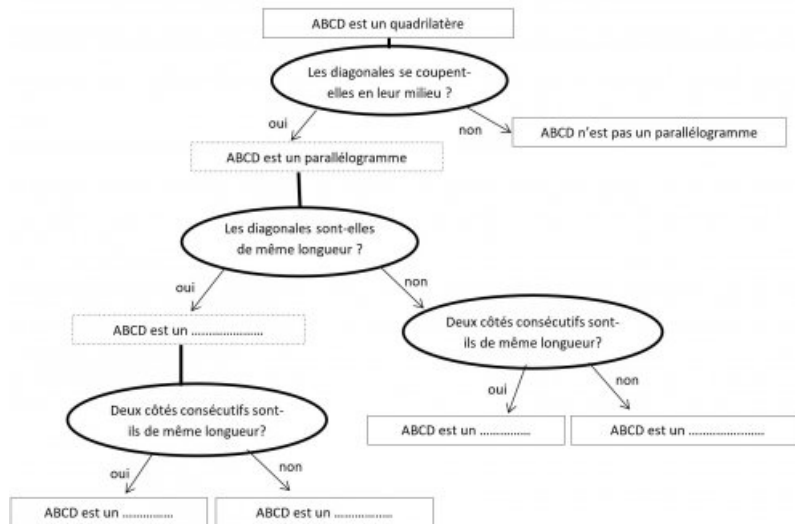
Ce document présente quelques commandes du langage Python et est illustrée par quelques missions à réaliser.

Le début de l'activité consolide la notion de variable autour du programme Scratch. L'univers Scratch permet de rassurer les élèves et leur donne un cadre assez simple du calcul des coordonnées du milieu d'un segment connaissant les coordonnées des extrémités. L'objectif sera progressivement d'introduire l'utilité du langage Python

en programmation.

Il semble préférable d'organiser la classe en flots afin que les échanges profitent aux élèves.

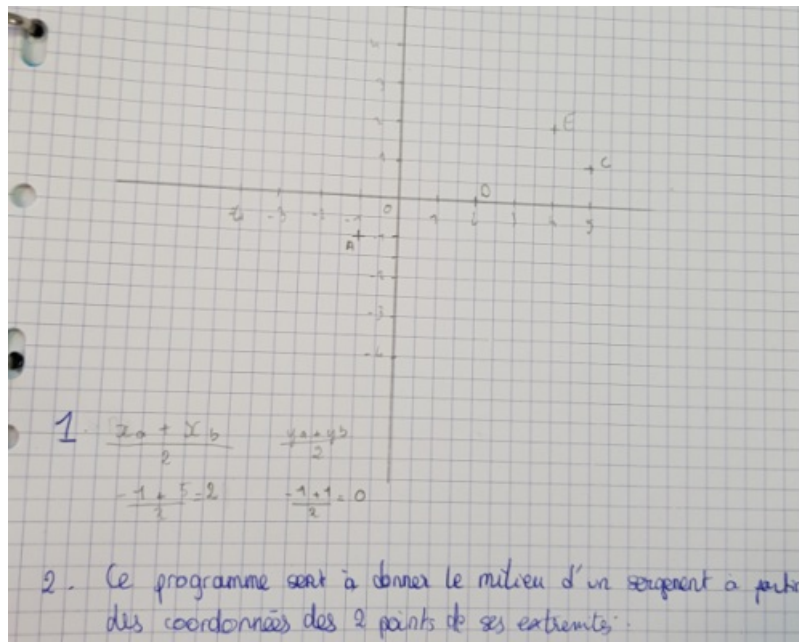
L'activité s'organise autour de cet algorithme qui n'est pas exploité dans sa totalité :



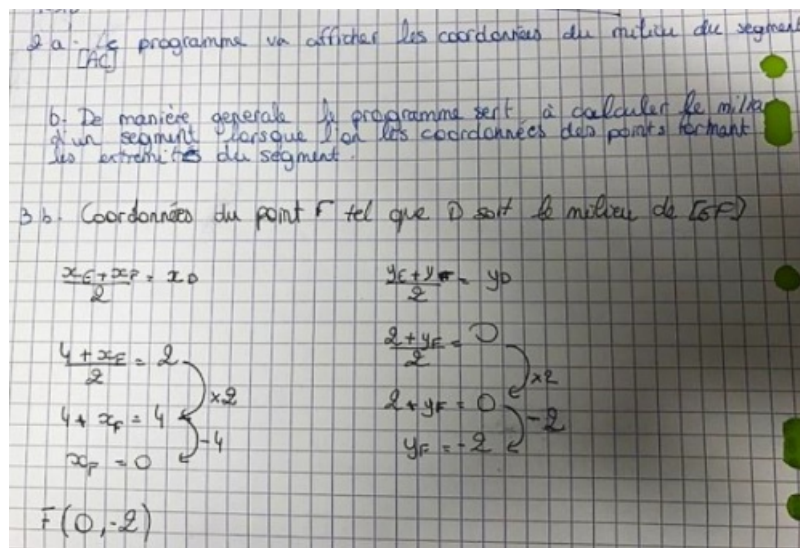
Cet algorithme permet d'identifier la nature d'un quadrilatère

Les questions 1, 2 et 3 amènent les élèves à retravailler la notion de milieu d'un segment connaissant les coordonnées des extrémités et s'approprier le premier programme en scratch. Les élèves ont assez bien réagi et l'aspect "débranché" de l'activité permet une mise au travail rapide et bien accompagnée. Cependant, ils s'attachent à faire une construction propre et cela prend du temps et retarde la réflexion sur l'essentiel de l'activité. On peut d'ailleurs demander de la réaliser sur le langage Geogebra afin d'en faciliter la construction.

La possibilité de manipuler sur le langage scratch avec le lien : [https://scratch.mit.edu/projects/204778540/](https://scratch.mit.edu/projects/204778540) favorise la compréhension des élèves. On constatera néanmoins que l'oubli de mettre "dire et attendre" pose problème si l'on veut le faire fonctionner devant les élèves, on voit seulement l'ordonnée du milieu !



Cette production dévoile le travail des élèves sur les premières questions du document géométrie repérée.



Cette production dévoile le travail des élèves sur les premières questions du document géométrie repérée.

La question 4 permet de montrer comment utiliser les fonctions en Python. Il s'appuie d'ailleurs sur le document d'accompagnement : <http://revue.sesamath.net/IMG/pdf/170516-algorithmique-lycee.pdf>. L'objectif est de supprimer les déclarations de variables. On peut d'ailleurs s'interroger du pourquoi cette partie est conservée : L'enseignant doit s'attacher à supprimer ces 5 lignes. On peut par exemple rentrer $\frac{1}{3}$ pour la valeur de a et observer que le programme annonce une erreur alors que si cela est rentré dans la fonction directement dans le shell, cela fonctionne. Par ailleurs, le choix de ces variables va d'ailleurs avoir une importance dans la suite. En effet, le test à effectuer dans la suite est entre le milieu du segment [AC] et celui du segment [DB]. Il convient donc d'expliquer aux élèves que ces variables ne sont pas liées aux points.

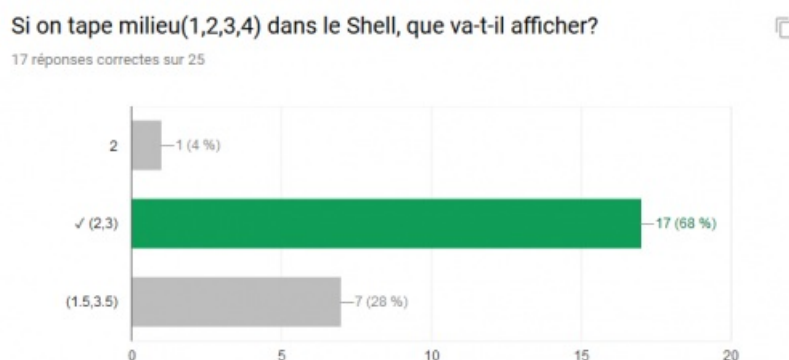
Les élèves ont rencontré de grandes difficultés pour comprendre la question 4)b). Contrairement à Scratch, on ne peut pas s'appuyer sur l'affichage de la valeur des variables pour expliquer le fonctionnement du programme. Rétrospectivement, je crois qu'il est préférable de définir la fonction avec a, b, c, et d puis demander de taper "milieu(xA, yA, xB, yB)". J'ai dû largement intervenir pour expliquer cette question. Le questionnaire Google Forms envoyé aux élèves en amont sous forme de pédagogie inversée,

Questionnaire prise en main Python (PDF de 88.9 ko)

Questionnaire réalisé avec Google forms pour préparer les élèves à la prise en main du langage Python

a constitué un appui certain pour s'assurer de la compréhension de tous. Les trois vidéos permettent de leur laisser une trace exploitable dans leurs travaux de révision.

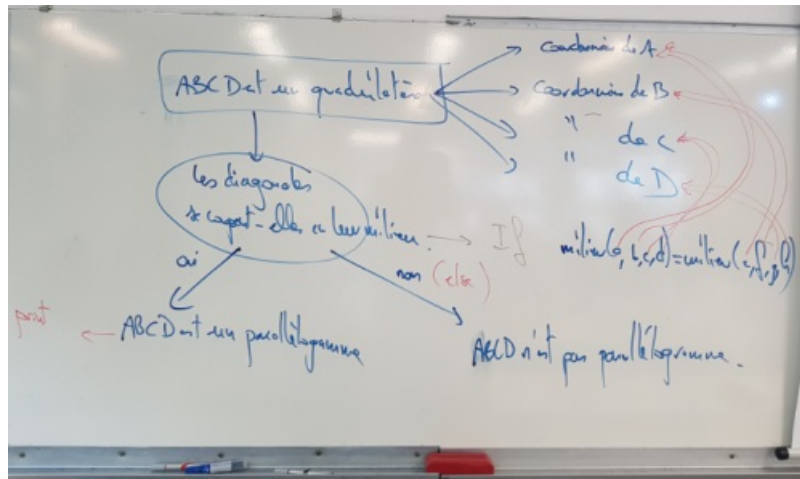
Voici les statistiques obtenus sur 25 élèves obtenus à la dernière question :



Statistiques des réponses au Google forms obtenus suite au questionnaire avec la classe.

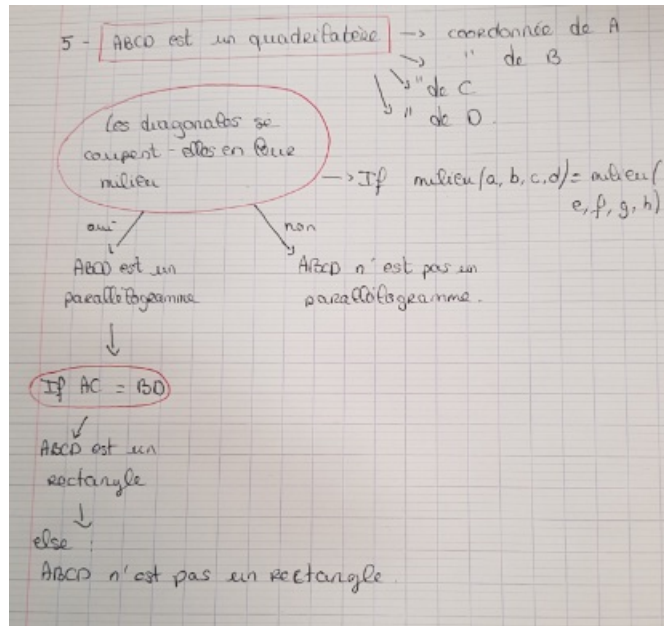
Les élèves intègrent facilement l'utilité d'une fonction et surtout la possibilité de la réintégrer dans un autre programme. Le fait de pouvoir ouvrir plusieurs fenêtres de programme et de faire des copier-coller constituent un atout pour mettre en œuvre ce type d'activité.

Arrivé à la question 5, il a été important de faire un bilan commun au tableau afin de faire comprendre la notion de variable modélisée par des flèches sur le tableau dans la photo ci-dessous et de préparer la boucle if, then, else :



Ce document témoigne de la synthèse des réflexions par les élèves.

Les élèves ont alors complété assez simplement l'algorithme pour vérifier que le quadrilatère est un rectangle :



Une fois ce travail réalisé et corrigé, les élèves ont programmé l'algorithme sous python. Certains élèves font de nombreuses confusions. Voici un élève qui comprend l'algorithme mais n'a pas assimilé la déclaration des variables sous Python. Il alterne xA et la variable a sans se servir de la notion de fonction.

```

#Calcul les coordonnées du milieu d'un segment

def milieu(xA,yA,xB,yB,xC,yC,xD,yD):
    return((xA+xB)/2,(yA+yB)/2,(xC+xD)/2,(yC+yD)/2 )

a=float(input('Quelle est l abscisse du point A'))
b=float(input('Quelle est l ordonnée du point A'))

c=float(input('Quelle est l abscisse du point B'))
d=float(input('Quelle est l ordonnée du point B'))

e=float(input('Quelle est l abscisse du point C'))
f=float(input('Quelle est l ordonnée du point C'))

g=float(input('Quelle est l abscisse du point D'))
h=float(input('Quelle est l ordonnée du point D'))
print(milieu(a,b,c,d,e,f,g,h))

if(xA+xB)/2,(yA+yB)/2==(xC+xD)/2,(yC+yD)/2:
    print('Le quadrilatère est un parallélogramme')
else:
    print('Le quadrilatère n est pas un parallélogramme')

```

Exemple de production d'élève en langage Python suite à la réflexion.

Il faut bien rajouter ici que la notion de fonction n'est pas naturelle. L'enseignant doit amener son utilité en montrant que l'on peut simplifier ce type de programme. Il semble donc nécessaire de convaincre par la pratique et l'étude de programme en "débranché".

A ce propos, le contrôle des programmes réalisés par les élèves peut s'avérer compliqué pour l'enseignant. Je conseille ici de s'appuyer sur quelques élèves ayant terminé l'activité ou possédant de bonnes compétences en programmation pour accompagner cette vérification. Ainsi, ces élèves pourront assister chaque groupe de la classe dans la découverte du langage.

La plupart de mes élèves ont atteint mes exigences et la possibilité d'ouvrir plusieurs programmes a facilité ce travail. Nous avons d'ailleurs débattu sur le double test à effectuer. Car, si il est fait séparément, on obtient seulement que les diagonales ont le même milieu ou qu'elles ont la même longueur. Cela permet de développer la compétence "Raisonnement" des élèves avec en particulier la différence entre le "OU" et le "ET". Certains élèves ont utilisé la commande "elif" et d'autres la commande "and".

```

from math import *
def milieu(xA,yA,xB,yB):
    return((xA+xB)/2,(yA+yB)/2)
def longueur(xA,yA,xB,yB):
    return sqrt(((xA-xB)**2+(yA-yB)**2))

a=float(input('quelle est l abscisse du point A'))
b=float(input('quelle est l ordonnée du point A'))

c=float(input('quelle est l abscisse du point B'))
d=float(input('quelle est l ordonnée du point B'))
print (milieu(a,b,c,d))

e=float(input('quelle est l abscisse du point C'))
f=float(input('quelle est l ordonnée du point C'))

g=float(input('quelle est l abscisse du point D'))
h=float(input('quelle est l ordonnée du point D'))
print(milieu(e,f,g,h))

if(milieu(a,b,c,d)==milieu(e,f,g,h)):
    print('ABCD est un parallélogramme')
else:
    print('ABCD n est pas un parallélogramme')
if(longueur(a,b,c,d)==longueur(e,f,g,h))and (milieu(a,b,c,d)==milieu(e,f,g,h)):
    print('ABCD est un rectangle')
else :
    print('ABCD n est pas un rectangle')

```

Deuxième production d'élève en langage Python suite à la synthèse formulée.

On peut néanmoins constater qu'ils éprouvent des difficultés à supprimer la déclaration des variables modélisant les coordonnées des points A, B, C et D. Il faut comprendre ici que les points A, B, C et D ne sont pas dans l'ordre du quadrilatère. L'élève en le faisant fonctionner a dû inverser les points B et C sinon, il obtient qu'un parallélogramme n'est pas un parallélogramme. Ceci peut d'ailleurs constituer une réelle difficulté de compréhension chez les élèves.

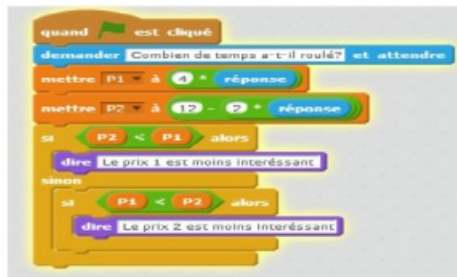
Même si l'objectif est atteint, on peut critiquer le test d'égalité établi. Il est facile de contourner le programme afin d'obtenir une égalité fautive qui induit que le quadrilatère n'est pas particulier alors qu'il l'est. On comprendra ici que l'objectif est de développer la compétence à utiliser les fonctions proposées dans Python et en conservant des nombres entiers comme coordonnées des points, cela ne posera pas de problème. Néanmoins, il n'est pas exclu de faire réagir les élèves sur ce problème en fin de séance.

On peut aussi retenir que retarder le passage sur l'ordinateur permet de mieux comprendre cette compréhension de l'outil fonction de Python qui est loin d'être naturel lors du passage à la programmation et qui reste difficile à enseigner aux élèves.

● Prolongements et évaluation :

Cet entraînement à la logique m'a permis de proposer en évaluation un exercice reposant sur ce principe. Ici, les élèves doivent comprendre l'intérêt du sinon dans le "si- alors-sinon". Je suis par contre encore resté dans l'univers de Scratch.

On considère l'algorithme suivant programmé sur scratch :



- 1) Expliquer ce qu'affiche l'algorithme si l'utilisateur répond 2 à la question « Combien de temps a-t-il roulé ? »
- 2) Expliquer ce qu'affiche l'algorithme si l'utilisateur répond 5 à la question « Combien de temps a-t-il roulé ? »
- 3) Quel est le problème de l'algorithme ?

Exercice proposé en évaluation en langage Scratch

● Documents de référence :

 [Géométrie repérée et algorithmique version élève](#) (PDF de 118.3 ko)

Document élève sur la notion de fonction en langage Python autour de la géométrie repérée.

 [Géométrie repérée et algorithmique version professeur](#) (PDF de 205.1 ko)

Document d'accompagnement pour le professeur sur la notion de fonction en langage Python autour de la géométrie repérée.