

Géométrie repérée et Algorithmique

LOIC CHAPELLIER

Ip2i, Jaunay-Marigny

Pour favoriser les échanges et l'avancé des élèves, il semble préférable d'avoir disposé la classe en îlots de 3 à 4 élèves.

Ce travail doit s'inscrire dans une progression structurée comprenant au préalable des exercices de maîtrise sur la formule de la longueur et du milieu d'un segment. Les commandes print et input ainsi que la bibliothèque math de Python doivent déjà être connues par les élèves.

Il sera intéressant de montrer que les outils de Geogebra tels que longueur et milieu utilisent ces deux formules et la fenêtre algèbre permet de les vérifier.

Voici un programme construit avec le logiciel Scratch:



On se place dans un repère orthonormé (O, I, J).

On pourra construire une figure que l'on complétera au fur et à mesure.

Premier temps (20 min) :

Intervention du professeur :

Le professeur présente le programme en scratch et invite les élèves à répondre aux premières questions.

Le professeur peut présenter l'objectif global de la séance et rappeler au préalable les propriétés des quadrilatères particuliers ainsi que les deux formules :

La distance AB entre deux points $A(x_A, y_A)$ et $B(x_B, y_B)$ dans un repère orthonormé (O, I, J) est donnée par la formule :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Dans un repère (O, I, J) , les coordonnées du milieu C d'un segment

avec $A(x_A, y_A)$ et $B(x_B, y_B)$ sont données par : $C\left(\frac{x_A + x_B}{2}, \frac{y_A + y_B}{2}\right)$

1. Que va-t-il afficher avec les points $A(-1;-1)$ et $C(5;1)$? Placer les points dans le repère construit.

Le professeur peut présenter un repère déjà construit, cela facilitera la construction de la figure. Il est aussi bien évidemment possible de demander aux élèves de construire la figure sur un logiciel de géométrie dynamique tel que Geogebra.

Il est possible de modifier le nom des points afin de « coller » davantage à la formule.

2. A quoi sert ce programme ?

Intervention du professeur :

Le professeur peut réaliser un premier bilan et inviter un élève à expliquer l'algorithme au tableau.

Le professeur peut faire fonctionner le programme avec scratch afin d'appuyer son explication et favoriser la compréhension des élèves.

3. On considère les points $E(4,2)$ et $D(2;0)$.
 - a) Placer les points dans le repère.
 - b) Déterminer les coordonnées du point F tel que D est le milieu de [EF].
 - c) Comment vérifier le résultat à l'aide du programme ?

Ce temps ne nécessite pas de bilan mais doit permettre au professeur en passant dans les rangs que l'utilité de l'algorithme est bien compris. L'enseignant peut d'ailleurs décider de ne pas traiter cette question selon le temps et la compréhension des élèves.

4. Voici le même algorithme programmé avec Python

```
def milieu(xA,yA,xB,yB):  
    return((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))
```

- a. Quel est l'avantage de la programmation Python ?
- b. Que se passe-t-il quand on tape la commande « milieu(a,b,c,d) » dans le programme.

Intervention du professeur :

Il est nécessaire d'insister sur l'intérêt de la programmation en Python. L'outil fonction permet de simplifier l'écriture du programme.

Il est nécessaire que l'enseignant explique aux élèves qu'avec le logiciel Python :

- *On peut ouvrir plusieurs fenêtres contrairement à scratch et faire des copiés collés.*
- *Une fonction peut être réutilisée dans un autre programme.*
- *On peut supprimer les commandes input et print afin de ne plus avoir la problématique de la déclaration de variable en tapant directement dans le shell.*

Un premier bilan écrit de tout cela peut être fait écrit par l'enseignant afin que les élèves conservent une trace de cette prise de conscience.

L'utilisation du shell peut constituer une difficulté pour les élèves. Elle n'est pas naturelle et doit être amenée par le professeur. Il est d'ailleurs possible de commencer à faire fonctionner le programme en « débranché ».

c. Utilisez cet algorithme afin d'en construire un qui permet de vérifier si un quadrilatère est un parallélogramme.

d. On considère les points $B(1;3)$ et $G(3;-3)$. Placer les points dans le repère dans le repère (O, I, J)

Utilisez l'algorithme pour vérifier que ABCG est un parallélogramme.

Ce temps ne nécessite pas de bilan mais doit permettre au professeur en passant dans les rangs que les fonctionnalités telles que le shell et l'outil fonction sont acquis. L'enseignant peut d'ailleurs décider de ne pas traiter cette question selon le temps et la compréhension des élèves.

Premier temps (40 min) :

Intervention du professeur :

Le professeur présente l'algorithme au tableau. Il est nécessaire de rappeler les propriétés des quadrilatères afin de les préparer à le compléter.

Il sera aussi nécessaire d'amener les élèves à leur faire comprendre l'intérêt du programme précédent dans cette démarche.

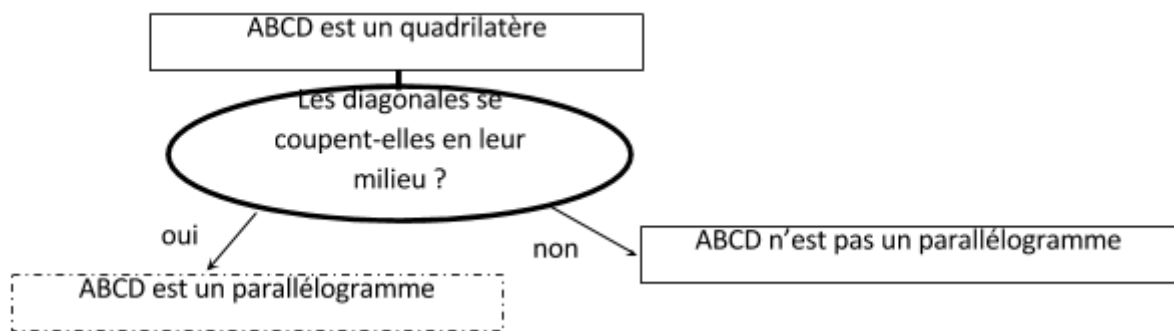
Il est aussi nécessaire d'expliquer ici les commandes `if`, `then`, `else`, `elif` de Python

Le professeur invite ensuite à poursuivre l'activité.

Le travail autour des instructions conditionnelles peut constituer un travail préalable qui favorisera la compréhension des élèves.

A ce stade, il est aussi possible d'expliquer ou de faire expliquer l'algorithme programmé en Python sur la longueur d'un segment. Après cette intervention, les élèves en groupe doivent pouvoir poursuivre en autonomie. L'enseignant peut d'ailleurs demander de terminer le programme en devoir ou en formative.

5. Cet algorithme peut être schématisé ainsi :



a. Cet algorithme peut être vu comme le point de départ d'un algorithme plus complet qui permettrait de vérifier qu'un quadrilatère est quelconque, parallélogramme ou rectangle.

Recopier le schéma et le compléter afin de pouvoir vérifier si le quadrilatère est rectangle.

b. Voici le programme suivant réalisé sous Python :

```
#Calcul les coordonnées du milieu d'un segment
from math import *

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(longueur(a,b,c,d))
```

A l'aide de cet algorithme, construire un algorithme qui permet de vérifier si un quadrilatère est un rectangle.

c. Utilisez cet algorithme pour vérifier si le quadrilatère ABCG est un rectangle.

Il sera nécessaire ici d'accompagner les élèves dans la double condition à vérifier dans le programme pour savoir si le quadrilatère est rectangle.

d. On considère les points $M(5;-4)$; $N(7;1)$; $K(2;3)$ et $L(0;-2)$.

Déterminer la nature du quadrilatère MNKL et vérifier le avec l'algorithme construit.

Intervention possible du professeur :

Le professeur présente le ou les programmes des élèves afin de montrer le fonctionnement ou d'y effectuer des modifications.

Ces deux dernières questions permettent pour l'enseignant de vérifier que le programme fonctionne car le quadrilatère est bien un rectangle. Il est possible d'inviter à vérifier le programme sur d'autres valeurs. L'enseignant peut d'ailleurs en proposer d'autres afin de vérifier si le programme teste bien l'égalité. Une réflexion autour de l'égalité de deux valeurs en programmation peut alors être menée et développera l'esprit critique des élèves.