

**Cadre de l'activité****Notions et contenus abordés**

Description du mouvement d'un système par celui d'un point.

Position. Trajectoire d'un point.

**Capacités exigibles. Activités expérimentales support de la formation**

Décrire le mouvement d'un système par celui d'un point et caractériser cette modélisation en termes de perte d'informations.

Caractériser différentes trajectoires.

**Capacité numérique**

Représenter les positions successives d'un système modélisé par un point lors d'une évolution unidimensionnelle ou bidimensionnelle à l'aide d'un langage de programmation.

**I. But de la séance**

Un centre de lancer de haches a ouvert récemment à Strasbourg. Le lien suivant vous permet de visualiser quelques [lancers](https://www.youtube.com/watch?time_continue=64&v=lbCPi_k7rF4) un peu particulier. [https://www.youtube.com/watch?time\_continue=64&v=lbCPi\_k7rF4]



Comment peut-on décrire qualitativement (puis quantitativement) la trajectoire de ces haches et d'autres objets plus généralement ?

**II. Mouvement d'un système. Mouvement du centre de gravité.****La trajectoire d'un objet**

Le dessin ci-dessus représente la **trajectoire** d'une hache, c'est à dire **la position de la hache prise à des intervalles de temps réguliers**.

⇒ Décrivez, avec vos mots, la trajectoire de la hache.

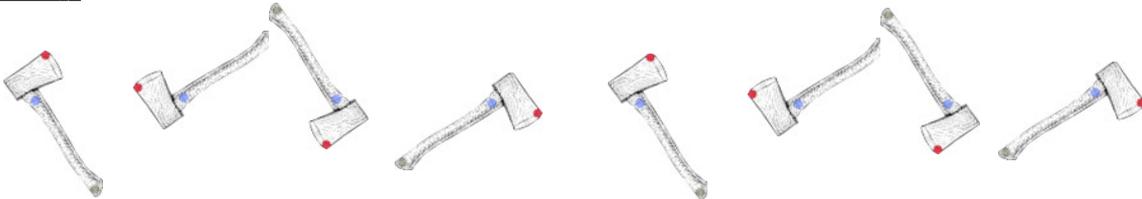
⇒ Reliez, à main levée, les différentes positions du point rouge de la hache pour obtenir la trajectoire du point rouge.

⇒ Faites de même avec les points bleus et gris.

⇒ Quel point possède la trajectoire qui se rapproche le plus de la trajectoire de la hache ?

**A retenir :**

Dans de nombreux cas ( et au moins en seconde ) nous étudierons systématiquement la trajectoire de ce point particulier\* plutôt que la trajectoire de l'objet lui-même. On appelle '**cinématique**' la branche de la physique qui étudie le mouvement. On appelle '**système**' l'objet dont on étudie la trajectoire. Enfin on appelle '**centre de gravité**' le point particulier\* dont le mouvement **modélise** celui du système.

**Perd-on des informations quand on étudie la trajectoire du centre de gravité à la place de la trajectoire du système ?**

Le dessin ci-dessus représente une **autre trajectoire** de hache (la hache tourne dans le sens rétrograde)

⇒ Tracez la trajectoire du centre de gravité.

⇒ Comparez cette trajectoire à celle du centre de gravité dans le premier lancer.

⇒ Que peut-on conclure ?

### III. Modéliser des trajectoires dans un langage de programmation (Python)

Comment peut-on simuler le déplacement d'un point dans le langage Python ?

Python permet de dessiner très rapidement de fenêtres graphiques dans lesquelles on peut placer des objets géométriques.

Le programme ci-dessous permet de créer une fenêtre graphique de 500 px sur 500 px et d'y placer un rond rouge en guise de système mécanique. Vous pouvez le télécharger [ici](http://www.lommele.com/activite_trajectoire.zip) [http://www.lommele.com/activite\_trajectoire.zip]

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # DLommele - lyceemangin
4  from tkinter import *
5  #####
6  # Fonctions #
7  #####
8  def Start():
9      pass
10 def Stop():
11     pass
12 #####
13 # PROGRAMME PRINCIPAL #
14 #####
15 t=0
16 x=250
17 y=20
18 racine=Tk()
19 racine.geometry("500x600")
20 racine.title("Positions")
21 fond=Canvas(racine, bg='white',width=500,height=500,bd=4)
22 fond.pack()
23 f=Frame(racine,width=500,height=100,bg='lightgrey')
24 f.pack()
25 debut=Button(f,text="Start",bg='lightgrey')
26 debut.pack()
27 fin=Button(f,text="Stop ",bg='lightgrey')
28 fin.pack()
29 quitter=Button(f,text="Quit ",bg='lightgrey', command=quit)
30 quitter.pack()
31 rond=fond.create_oval(x,y,x+10,y+10,fill='green')
32 racine.mainloop()

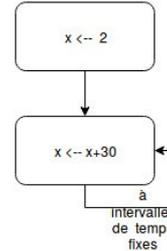
```

Pour l'instant les boutons ne sont pas actifs. Le point ne se déplace pas (encore). Mais nous pouvons déjà modifier sa position 'initiale' en modifiant les lignes 16 et 17.   
 ⇒ Remplacez la ligne 17 par 'y=250' et regardez ce qui se passe.   
 ⇒ Remplacez la ligne 16 par 'x=400' et regardez ce qui se passe.   
 ⇒ Modifiez les lignes 16 et 17 pour placer le point en bas à gauche.

Si nous pouvons déplacer nous-même le point, étape par étape, nous pouvons aussi automatiser ce déplacement.

Il suffit d'affecter d'autres valeurs à x (et à y) à des intervalles de temps réguliers.

⇒ En interprétant l'algorithme ci-dessous, donnez les différentes abscisses x du point.



Le programme ci-dessous déplace le point rouge selon l'algorithme précédent.

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # DLommele - lyceemangin
4  from tkinter import *
5  #####
6  # Fonctions #
7  #####
8  def Start():
9      global x,y,t
10
11         x=x+30
12
13         fond.coords(rond,x,y,x+10,y+10)
14
15         racine.after(500,Start)
16 def Stop():
17     pass
18 #####
19 # PROGRAMME PRINCIPAL #
20 #####
21 t=0
22 x=2
23 y=250
24 racine=Tk()
25 racine.geometry("500x600")
26 racine.title("Positions")
27 fond=Canvas(racine, bg='white',width=500,height=500,bd=4)
28 fond.pack()
29 f=Frame(racine,width=500,height=100,bg='lightgrey')
30 f.pack()
31 debut=Button(f,text="Start",bg='lightgrey',command=Start)
32 debut.pack()
33 fin=Button(f,text="Stop ",bg='lightgrey')
34 fin.pack()
35 quitter=Button(f,text="Quit ",bg='lightgrey', command=quit)
36 quitter.pack()
37 rond=fond.create_oval(x,y,x+10,y+10,fill='green')
38 racine.mainloop()

```

On constate que l'algorithme a été 'traduit' en langage Python de la ligne 8 à la ligne 15. La ligne 9 indique les variables qui seront modifiées dans le reste du programme. La ligne 11 modifie la valeur de l'abscisse. La ligne 13 redessine le rond rouge à ses nouvelles coordonnées. La ligne 15 relance la fonction Start toutes les 500 ms.

⇒ Lancez le programme n°2 que vous pouvez le télécharger [ici](#).

#### Comment garder des traces des différentes positions du rond rouge ?

On apporte à la fonction Start() quelques modifications :

```

8  def Start():
9      global x,y,t
10         fond.create_oval(x+4,y+4,x+6,y+6,fill='black')
11         x=x+30
12         t=t+0.5
13         fond.coords(rond,x,y,x+10,y+10)
14         a="t=\n"+str(t)+"s"
15         fond.create_text(x,y+30,text=a,width=25)
16         racine.after(500,Start)

```

La ligne 10 dessine un point noir à la dernière position du cercle.

La ligne 12 permet de modifier les valeurs de t, le temps.

Les lignes 14 et 15 permettent d'afficher les valeurs de t sous le point correspondant.

⇒ Lancez le programme n°3 que vous pouvez le télécharger [ici](#).

⇒ Modifiez le programme n°3 pour que le rond rouge se déplace de la droite vers la gauche toutes les 800 ms.

### IV. Caractériser des trajectoires différentes

Si la trajectoire du point considéré dans l'étude du mouvement est une droite, on dit que le mouvement est rectiligne.

Si l'écart entre les points reste le même pour des intervalles de temps fixes, on dit que le mouvement est uniforme.

⇒ Lancez les programmes [n°4 à n°7](#) et décrivez le mouvement du point en précisant s'il est rectiligne et/ou uniforme.

Si vous n'êtes pas en mesure d'exécuter les programmes, vous pouvez trouver les trajectoires correspondantes ci-dessous.

## V. Les différentes trajectoires

