Mouvement dans un champ de pesanteur

Niveau (Thèmes)	Seconde : Mouvement et interactions
Introduction	À partir de l'enregistrement vidéo du mouvement d'un objet dans le champ de pesanteur (chute libre verticale, avec vitesse initiale, dans l'air ou dans un fluide), les élèves vont devoir pointer les positions successives de l'objet puis extraire ces données et, à l'aide d'un langage de programmation, représenter la trajectoire et calculer puis représenter les vecteurs vitesse instantanée en chacun des points.
Type d'activité	Activité expérimentale (réaliser le pointage) et numérique (extraire les positions et calculer les coordonnées des vecteurs vitesse instantanée en chaque point).
Compétences	ANALYSER : - Planifier des tâches - Relier différents types de représentations RÉALISER - Réaliser une partie d'un programme sous python VALIDER - Discuter de la validité d'une modélisation - Proposer des améliorations de la démarche
CRCN - Compétences Num.	Distinguer une simulation ou une modélisation de la réalité lors du traitement des informations. Préciser le contexte associé aux résultats obtenus et ses conséquences sur leur interprétation. Identifier la nature des modèles employés et leurs limites de validité.
Notions et contenus du programme	Vecteur vitesse d'un point. Réaliser et exploiter une vidéo d'un système en mouvement et représenter des vecteurs vitesse. Capacité numérique : représenter des vecteurs vitesse d'un système modélisé par un point lors d'un mouvement à l'aide d'un langage de programmation.
Objectif(s) pédagogique(s)	Comprendre la notion de vitesse instantanée. Comprendre le fonctionnement d'un logiciel de pointage et ses limites. Maîtriser la notion de vecteur vitesse.
Objectifs disciplinaires et/ou transversaux	
Description succincte de l'activité	Dans une première séance, les élèves réalisent l'enregistrement vidéo d'un mouvement dans le champ de pesanteur, par exemple un lancer dans une direction quelconque, ou un lâcher.. À l'aide d'un logiciel de pointage, ils enregistrent les positions successives de l'objet et les exportent afin d'être utilisées par la suite. Ces données constituent la matière à analyser à l'aide du langage de programmation. Il s'agira de calculer les coordonnées des vecteurs vitesse en chaque point et de les représenter.
Découpage temporel de la séquence	La séquence se déroule selon l'organisation : - introduction des notions théoriques : vitesse, vecteur déplacement, vecteur vitesse (classe entière) - réalisation de la vidéo + pointage (TP) - tracé des courbes sur papier millimétré et calcul des vecteurs vitesse (classe entière + maison) - exploitation numérique, modélisation (TP)
Pré-requis	Lire et modifier un langage python (voir activités "simulation des positions d'un point en mouvement") Notion de vecteur
Outils numériques utilisés/Matériel	Salle équipée d'ordinateurs sur lesquels sera installée la suite logicielle "edupython" ou autres
Gestion du groupe Durée estimée	2 séances de groupe de 1,5h + 2 h classe entière pour cours sur vitesse, vecteur déplacement...

ACTIVITÉ N°2 : EXPLOITATION NUMÉRIQUE DE L'ENREGISTREMENT VIDÉO DU LANCER D'UNE BALLE

L'objectif de cette séance est de procéder à l'analyse du mouvement enregistré et, en particulier, d'étudier la vitesse de la balle au cours de son mouvement.

Partie 1 : sur feuille, construction de la trajectoire de la balle.
À partir du fichier « lancer_groupe-xx.csv », qui contient les coordonnées des positions de la balle en fonction du temps, construire la trajectoire de la balle dans le repère OXY défini lors de l'acquisition réalisée à partir de l'enregistrement vidéo.

Partie 2 : sur feuille, calcul des coordonnées du vecteur vitesse en un point de la trajectoire.
1. On note Xi et Yi les coordonnées de la position *i* de la balle. Donner l'expression des coordonnées du vecteur vitesse $\vec{V}$ de la balle à la position i.
2. Calculer les coordonnées du vecteur vitesse de la balle lorsqu'elle atteint sa position i=4.

Partie 3 : programmation.
Vous disposez d'un programme « etude-mouvement_balle.py » qui, en l'état, permet d'extraire les données acquises lors du pointage sous la forme :

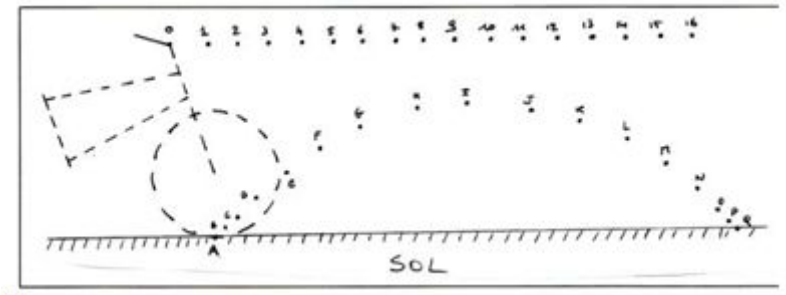
- d'une liste « T » qui contient les valeurs des instants correspondant aux positions de la balle,
- d'une liste « X » qui contient les valeurs des abscisses des positions de la balles,
- d'une liste « Y » qui contient les valeurs des ordonnées des positions de la balle.

1. Écrire en langage naturel l'algorithme qui permettrait de calculer les coordonnées du vecteur vitesse en chacun des points de la trajectoire.
2. À partir de la réponse précédente, créer et insérer à la ligne 53 une boucle qui permette de :
   - calculer les coordonnées du vecteur vitesse pour chaque position de la balle,
   - remplir au fur et à mesure les listes vitesses_x et vitesses_y.

FICHE OUTIL 1 : VECTEUR VITESSE

TRACER LE VECTEUR VITESSE « À LA MAIN »

Pour tracer le vecteur vitesse sur la trajectoire d'un objet, il faut avoir accès à sa trajectoire spatiale et temporelle : Par exemple, l'ensemble des points, pris à intervalle de temps constant.



La durée entre chaque point est 100 ms.
L'échelle est de 1 cm pour 15 cm

La valeur du vecteur vitesse au point H, sera la distance entre les points immédiatement à côté G et I divisés par le temps de parcours.

=> À FAIRE : calculer la valeur du vecteur vitesse au point H.

La direction du vecteur vitesse est donnée par la tangente à la trajectoire au point H. **Le sens** se déduit de l'énoncé.

=> À FAIRE : tracer le vecteur vitesse V_H avec comme échelle 1cm <-> 1 m/s

LE VECTEUR VITESSE POUR UN ORDINATEUR

Si on appelle « i » le « i-ième » point, on calcule les coordonnées du vecteur vitesse ainsi :

- coordonnée du vecteur vitesse selon l'axe des abscisses V_x :

$$V_X = \frac{(abscisse\ du\ (i+1)ème\ point\ -\ abscisse\ du\ (i-1)ème\ point)}{(temps\ du\ (i+1)ème\ point - temps\ du\ (i-1)ème)}$$

- coordonnée du vecteur vitesse selon l'axe des ordonnées V_y *:*

$$V_Y = \frac{(ordonnée\ du\ (i+1)ème\ point\ -\ ordonnée\ du\ (i-1)ème\ point)}{(temps\ du\ (i+1)ème\ point - temps\ du\ (i-1)ème)}$$

En python : avec les listes "abscisses" et "temps" (à adapter).

$$V_X = \frac{(abscisses[n+1] - abscisses[n-1])}{(temps[n+1] - temps[n-1])}$$

FICHE OUTIL 2 : NOTIONS DE LANGAGE PYTHON

LES LISTES EN PYTHON

Une **liste** est un objet informatique que l'on peut aussi appeler **tableau**. Elle est définie avec des '['.
Elle peut contenir tout type d'objets python : chiffres, nombres, lettres, etc.

Ajouter une valeur à une liste

Pour ajouter une valeur à une liste, à la fin, on utilise le mot clé **« append ».**
Exemple :
Liste = [1,2,3,4]
Puis
Liste.append(142) <= c'est la commande « append »
Donnera la liste : [1,2,3,4,142]

Connaître le nombre d'objets contenus dans une liste

On utilise le mot clé "len".
Exemple : len([1,2,3,4,5,6,'a']) renverra le chiffre 7 car la liste contient 7 objets.

Récupérer la valeur d'un item de liste en fonction de sa position

Si on veut récupérer l'item numéroté 3 : liste[3]
Si on veut récupérer l'item défini par sa position "i" : liste[i]

FICHE D'UTILISATION DU MODULE PYTHON MATPLOTLIB

Pré-requis :
Une liste python contenant les abscisses et une autre contenant les ordonnées
Par exemple :
Liste des abscisses : *abscisses = [1,2,3,4,5]*

Liste des ordonnées : *ordonnees = [10,20,30,40,50]*

Pour tracer le graphique **ordonnees = f(abscisses),** il suffit d'écrire les lignes suivantes :

```
import matplotlib.pyplot as plt                 #importe le module nécessaire pour tracer un graphe
axes = plt.gca()                                # crée un système d'axes
plt.plot(abscisses, ordonnees, '+')             # trace les points dont les abscisses et ordonnées sont
dans les listes
plt.title('trajectoire d\'une balle lancée')    #mets un titre
plt.savefig('trajectoire.png')                    #sauvegarde le graphe sous le nom "trajectoire.png"
plt.show()                                      #montre le graphique
plt.xlabel('x')                                 #rajouter un label sur l'axe des abscisses
plt.ylabel('y')
plt.close()                                     #referme le graphe
                                                #quand on cliquera sur la croix de fermeture.
```

Le programme python support (à compléter):

```
#importer les bibliothèques
import matplotlib.pyplot as plt
import csv

#ouvrir le fichier contenant les données
f = open('lancer_professeur.csv') #doit se trouver dans le même dossier que le programme python
fichier = csv.reader(f, delimiter=';')

#création de listes contenant les données récoltées : instant du pointage, abscisse de la balle à cet instant,
#ordonnée de la balle à ce même instant

T = [ ]
X = [ ]
Y =[ ]

for ligne in fichier:
        if ligne[0] == "" :
                a=1
        else:
                t = float(ligne[0].replace(',','.')) #on remplace la virgule par un point dans les données pour qu'elles
                soient bien lues comme des nombres
                x = float(ligne[1].replace(',','.'))
                y = float(ligne[3].replace(',','.'))
                T.append(t) #la commande append permet d'ajouter la valeur t calculée (la ligne 19) à la liste T
                #définie plus haut (ligne 11)
                X.append(x)
                Y.append(y)
print(T)
print(X)
print(Y)
n=len(T)
min_x = min(X)
max_x = max(X)
min_y = min(Y)
max_y = max(Y)
print(min_x)
print(max_x)
print(min_y)
print(max_y)
print(n)

#########################################################################
#########################################################################

#calcul coordonnées vecteur vitesse
#on commence par définir les coordonnées du vecteur vitesse pour chaque point comme des listes
vitesses_x = [ ]
vitesses_y = [ ]

#############################################################
#les élèves doivent écrire cette partie du programme-----*
#calculer les coordonnées du vecteur vitesse




#############################################################
axes = plt.gca()
plt.axis('equal')
plt.title("trajectoire de la balle")
```

```
axes.set_xlim(min_x, max_x)
axes.set_ylim(min_y, max_y)
plt.xlabel("abscisses")
plt.ylabel("ordonnées")
#########################################################################
#tracer la trajectoire et les vecteurs vitesse
plt.plot(X, Y, "+")
for i in range(1,n-1):
        plt.quiver(X[i], Y[i], vitesses_x[i-1], vitesses_y[i-1], angles='xy')
plt.savefig('trajectoire.png') #enregistre le graphique dans un fichier image
#########################################################################
plt.show()
plt.close()
```

Corrigé pour les enseignant.e.s

Partie 2

Les élèves ont vu en classe que le vecteur vitesse moyenne d'un point est défini à partir du vecteur déplacement entre les positions antérieure et postérieure.

Ils savent écrire $v_x(M_i) = [X(M_{i+1}) - X(M_{i-1})] / [t(M_{i+1}) - t(M_{i-1})]$ et $v_y(M_i) = [Y(M_{i+1}) - Y(M_{i-1})] / [t(M_{i+1}) - t(M_{i-1})]$. La question 1 est donc une question de cours.

Pour la question 2, ils font l'application pour le point M_4.

Partie 3

ATTENTION. L'export des données depuis le logiciel de pointage peut donner un tableau avec des lignes vides et des lignes qui contiennent des lettres (titres de colonnes). Le programme fourni parvient à tenir compte des lignes vides mais il est nécessaire de supprimer les lignes contenant des lettres, sans quoi des erreurs apparaissent au cours de la compilation.

question 1 : langage naturel

Une réponse de ce type est attendue :

```
pour i variant entre 1 et n-1:
calculer v_x_i
ajouter v_x_i à la liste vitesses_x
calculer v_y_i
ajouter v_y_i à la liste vitesses_y
```

question 2 : code demandé aux élèves

```
############################################################
#les élèves doivent écrire cette partie du programme-----*
#calculer les coordonnées du vecteur vitesse
for i in range(1,n-1):
    v_x_i=(X[i+1]-X[i-1])/(T[i+1]-T[i-1])
    vitesses_x.append(v_x_i)
    v_y_i=(Y[i+1]-Y[i-1])/(T[i+1]-T[i-1])
    vitesses_y.append(v_y_i)
############################################################
```

Retour d'expérience :

Les plus-value pédagogiques (enseignants/élèves) :

Les freins :

L'activité s'est avérée difficile pour les élèves sur les points suivants :

- les formules avec indice posent des difficultés de compréhension de la notion d'indice ;
- le passage de la formule "mathématique écrite" à la formule "traduite en langage Python" avec l'appel à un élément de la liste : commande X[i] pour appeler l'élément i de la liste X ;
- l'utilisation de la boucle for pour traduire la notion de procédure itérative.

Les leviers :

Cette activité est la troisième consacrée à la programmation et les élèves ont commencé à acquérir des automatismes, même si les difficultés évoquées demeurent.
Le fait de lire le programme avec eux et d'en faire ressortir la structure les aide à identifier ce qu'ils doivent faire.
L'explicitation de la distinction entre expression mathématique et traduction en langage Python permet aux élèves de comprendre où se fait le lien entre ce dont ils ont l'habitude (écrire une équation) et ce qu'ils apprennent à faire (écrire en langage informatique).

Les pistes pour aller plus loin ou généraliser la démarche :

Production d’élèves :

à venir