

Simulation des positions d'un point en mouvement

Niveau (Thèmes)	Seconde (nouveau programme) : Mouvement et interactions
Introduction	A travers une série de trois activités, les élèves vont revoir les notions de mouvement et de trajectoire, découvrir un nouveau langage de programmation ("Python") en utilisant leurs connaissances sur le langage Scratch travaillé au collège.
Type d'activité	Activités documentaires (lire des programme sous Scratch et Python). Activités de programmation sous python.
Compétences	<p>S'APPROPRIER :</p> <ul style="list-style-type: none"> - Relier entre elles des informations relatives à deux langages de programmation <p>ANALYSER :</p> <ul style="list-style-type: none"> - Relier différents types de représentation - Repérer ou sélectionner des informations utiles <p>RÉALISER</p> <ul style="list-style-type: none"> - Utiliser un langage de programmation : python <p>VALIDER</p> <ul style="list-style-type: none"> - Discuter de la validité d'une information <p>COMMUNIQUER</p> <ul style="list-style-type: none"> - Utiliser un vocabulaire adapté
CRCN - Compétences Num.	
Notions et contenus du programme	<p>Description du mouvement d'un système par celui d'un point. Caractériser différentes trajectoires. 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.</p>
Objectif(s) pédagogique(s)	<p>Maîtriser ou découvrir les notions de mouvement rectiligne uniforme, accéléré, ralenti À partir d'un programme Scratch, découvrir le langage Python en repérant les instructions communes ou différentes aux deux langages. Modifier un programme sous Python, utiliser les modules "turtle" et "matplotlib". Découvrir la notion de variable.</p>
Objectifs disciplinaires et/ou transversaux	
Description succincte de l'activité	<p>Les trois activités prennent place dans une séquence sur l'étude du mouvement en classe de seconde. Elles consistent successivement à :</p> <p>Activité 1 : Le professeur fournit un programme écrit sous scratch qui permet de simuler un mouvement rectiligne uniforme. Les élèves doivent alors prévoir le résultat de l'exécution du programme puis modifier le programme pour simuler des mouvements rectilignes accéléré et ralenti.</p> <p>Activité 2 : Le professeur fournit un programme python conçu à partir d'un module graphique du langage Python, le module <i>turtle</i>, qui permet de simuler un mouvement rectiligne uniforme. Les élèves doivent identifier les points communs entre ce programme et le programme scratch de l'activité 1.</p> <p>Puis les élèves doivent écrire le programme sous Python en utilisant le module <i>turtle</i> afin de simuler les trois types de mouvements rectilignes : uniforme, accéléré et ralenti. Ils doivent pour cela utiliser une variable "vitesse" associée à chaque mouvement.</p> <p>Activité 3 : Le professeur fournit aux élèves un programme similaire à celui de l'activité 2 mais en utilisant la librairie <i>matplotlib</i> qui offre la possibilité de tracer des graphiques. Les élèves pourront alors utiliser le programme pour construire la trajectoire et les vecteurs vitesse en chacun des points.</p>

	<p>Turtle est un module graphique du langage de programmation python. Il permet de visualiser des tracés (ici des points). Il contient une bibliothèque de fonctions (mot) mises à disposition.</p> <p>Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques.</p>
Découpage temporel de la séquence	<p>La séance se déroule selon l'organisation :</p> <ul style="list-style-type: none"> - une séance en salle équipée d'ordinateurs permettant de réaliser les activités 1 et 2. (par groupe de 2 élèves) - une séance en salle équipée d'ordinateurs permet de réaliser l'activité 3 (par groupe de 2 élèves)
Pré-requis	Avoir déjà utilisé Scratch au collège.
Outils numériques utilisés/Matériel	<p>Ordinateurs équipés des logiciels Scratch 2 ou 3 et Edupython ou Python 3.7.2</p> <p>https://scratch.mit.edu/download</p> <p>https://edupython.tuxfamily.org/</p> <p>https://www.python.org/downloads/</p> <p>Télécharger la librairie matplotlib (https://www.science-emergence.com/Articles/Tutoriel-Matplotlib/)</p>
Gestion du groupe Durée estimée	Travail en binôme : 2 x 1h30

Énoncés à destination des élèves

Simuler des mouvements

Au collège, vous avez appris à reconnaître les mouvements (trajectoires et vitesses) des objets. Dans cette activité, nous créerons à l'aide de programmes informatiques la représentation de mouvements rectilignes : uniforme, accéléré et ralenti.

Activité 1 : Utilisation du langage scratch

1 / Analyser un programme :

- Lire le programme ci-contre écrit avec le langage scratch.
- Dessiner le résultat de ce programme dans le cadre ci-dessous :



2 / Ouvrir le document : "act1_mouvement_uniforme.sb2" et le télécharger.

3 / Mouvement accéléré :

- Créer un nouveau "lutin" : Vous pouvez utiliser l'outil : "dupliquer".
- Le lutin sera placé en dessous du premier lutin sur la route.
- Modifier le programme pour que le lutin représente un mouvement accéléré.

4 / Mouvement ralenti :

- Créer un nouveau "lutin" : Vous pouvez utiliser l'outil : "dupliquer".
- Le lutin sera placé en dessous du deuxième lutin sur la route.
- Modifier le programme pour que le lutin représente un mouvement ralenti.

5 / Exécuter votre programme :

- Faire vérifier votre travail.
- Dans l'ENT, enregistrer votre fichier, sous le nom : "mouvement_uniforme_NOMS.sb2".

6 / Si vous avez du temps, modifier la couleur des points, la forme des lutins, ...

Activité 2 : Utilisation du module Turtle sous Python

Le même programme est écrit dans le langage Python.
Stylo est le nom du stylo n°1

1 / Analyser un programme

- Retrouver les correspondances entre les instructions de ce programme et les instructions écrites sous Scratch.
- Compléter le tableau ci-dessous :

```
from turtle import *  
  
#variable  
stylo = Pen()  
vitessestylo = 50 #en m/s  
dt = 1 #en s  
  
stylo.up()  
stylo.goto(-240,0)  
  
for i in range(8):  
    stylo.dot()  
    stylo.forward(vitessestylo*dt)
```

Commande scratch	Commande Python

2 / Ouvrir le logiciel edupython (ordinateur / Appli(V:) / Edupython / IDLE.exe)

- Recopier ce programme puis l'exécuter. (Attention à bien respecter les majuscules et le minuscules)
- Faire vérifier votre travail.
- Dans l'ENT, enregistrer votre fichier sous le nom act2_mouvement_rectiligne_uniforme_NOMS.py

3 / Mouvement rectiligne accéléré :

- Modifier le programme pour représenter un mouvement accéléré.
- Exécuter votre programme
- Dans l'ENT, enregistrer votre fichier sous le nom act2_mouvement_rectiligne_accélééré_NOMS.py

4 / Mouvement rectiligne ralenti :

- Modifier le programme pour représenter un mouvement ralenti.
- Exécuter votre programme.
- Dans l'ENT, enregistrer votre fichier sous le nom act2_mouvement_rectiligne_ralenti_NOMS.py

Activité 3 : Utilisation du module matplotlib sous python

Il est possible de se passer du module *turtle* et d'utiliser à la place un module qui permet de construire des graphiques. Le programme est alors modifié comme indiqué dans le fichier "mouvement_rectiligne_unif_act3.py".

1/ Exécuter le programme. À quoi correspondent les flèches sur le graphique ?

2/ Modifier la valeur de la vitesse puis exécuter à nouveau le programme. Faire les adaptations éventuelles pour que le tracé soit convenable.

3/ Enregistrer le programme sous "mouvement_rectiligne_accel_NOMS.py" et procéder aux modifications nécessaires pour représenter un mouvement rectiligne accéléré.

4/ Enregistrer le programme sous "mouvement_rectiligne_ral_NOMS.py" et procéder aux modifications nécessaires pour représenter un mouvement rectiligne ralenti.

Le programme :

```
#Ce programme permet de suivre les positions d'un point ayant un
#mouvement rectiligne en utilisant le module 'matplotlib'

#il faut d'abord importer le module auquel on fera référence par 'plt'

import matplotlib.pyplot as plt

#il faut ensuite définir les conditions initiales (position et vitesse à l'instant t=0 s)
x_0 = 0 #en mètre
y_0 = 5
v_0 = 5 #en m/s

#définir la durée de l'étude et le nombre de positions que l'on souhaite afficher
T = 10 #en seconde
n = 10

#paramétrer l'itération : calcul pas à pas des positions successives du point

dt = T/n #durée du "pas"
i = 1 #démarre l'itération

#définition des variables de position et de vitesse comme des listes
#[x_0, x_1, x_2 ...] qu'il va falloir construire pas à pas
X = [x_0]
Y = [y_0]
valeur_vitesse = [v_0]

#boucle itérative
while i <= n:
    x_i = x_0 + v_0*dt
    x_0 = x_i
    X.append(x_i) #la commande nom.append permet d'ajouter un élément à la liste 'nom'
    Y.append(y_0)
    valeur_vitesse.append(v_0)
    i += 1      #! cette commande se lit i=i+1

#tracer les positions successives ainsi que le vecteur vitesse du point
#pour chacune de ces positions

axes = plt.gca()
axes.set_xlim(0, 50) #possibilité de définir les limites des axes.
axes.set_ylim(0, 6)
#boucle qui permet de construire le graphique point par point
plt.title("Mouvement rectiligne uniforme")
plt.xlabel("abscisses")
plt.ylabel("ordonnées")
for j in range(n):
    plt.plot(X[j], Y[j], ".")
    plt.pause(0.5)
#boucle qui permet d'ajouter les vecteurs vitesse position par position
for j in range(n):
    plt.quiver(X[j], Y[j], valeur_vitesse[j], 0, scale = 80)
```

```
plt.pause(0.5)
```

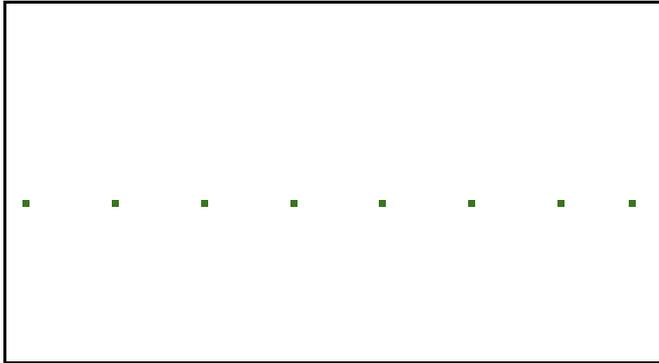
```
plt.savefig('mouvement_rectiligne_uniforme.png') # enregistre le graphique dans un fichier image  
plt.show()  
plt.close()
```

Corrigé pour les enseignant.e.s :

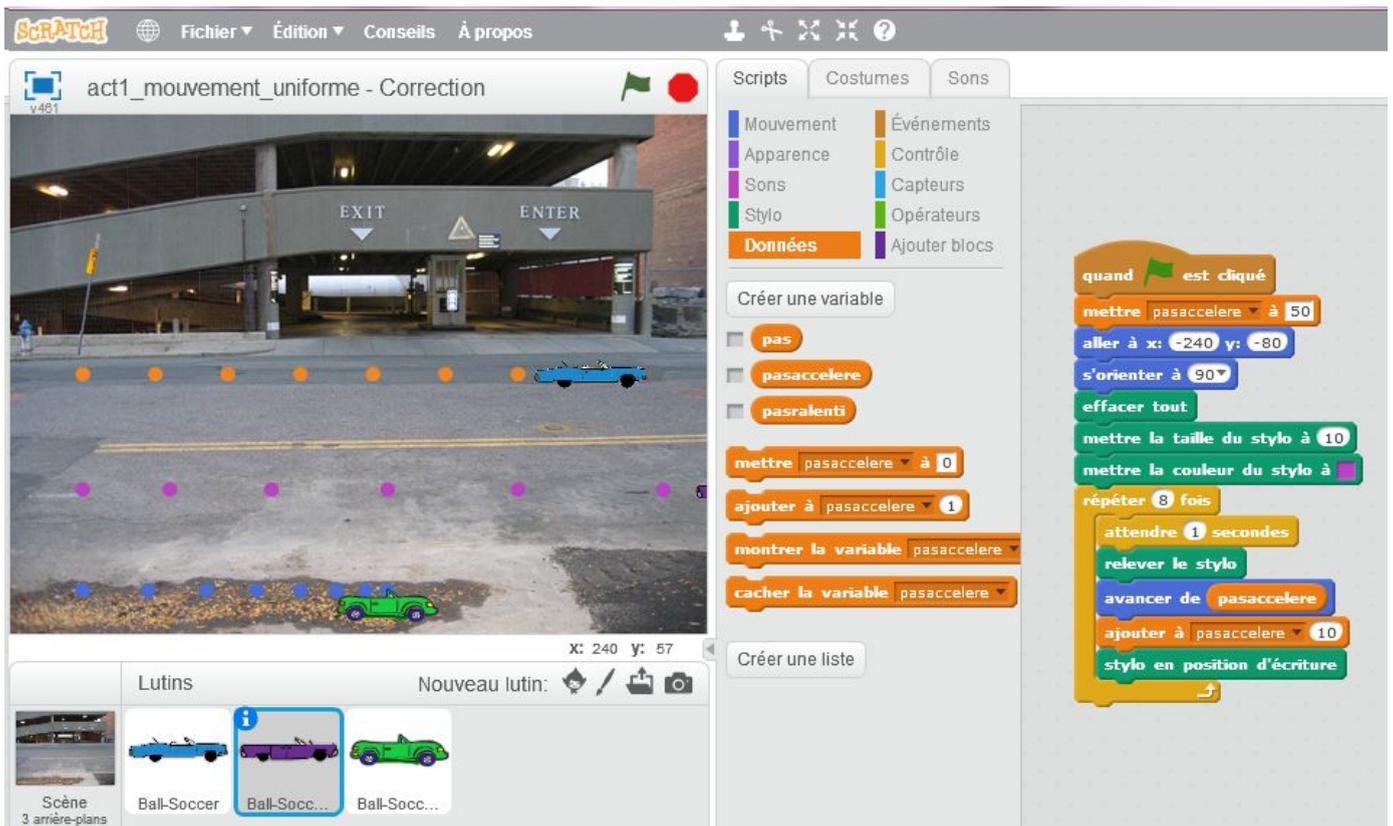
Activité 1 :

1 / Analyser un programme :

- Lire le programme ci-contre écrit avec le langage scratch.
- Dessiner le résultat de ce programme dans le cadre ci-dessous



```
quand est cliqué
mettre pas à 50
aller à x: -200 y: -90
s'orienter à 90
effacer tout
mettre la taille du stylo à 10
mettre la couleur du stylo à
répéter 8 fois
  attendre 1 secondes
  relever le stylo
  avancer de pas
  stylo en position d'écriture
```



The screenshot shows the Scratch IDE interface. The stage displays a soccer field with a blue car and a green car. The Scripts area shows a script with a 'pas' variable and a loop of 8 iterations. The loop contains: 'attendre 1 secondes', 'relever le stylo', 'avancer de pas', and 'stylo en position d'écriture'.

Copie d'écran : Résultat final à obtenir pour l'activité 1

Aide : il faut dupliquer les lutins puis créer une nouvelle variable (ex : pasaccelere). Il faut alors créer une variable pour ce lutin uniquement.

Activité 2 :

1 / Analyser un programme

- Retrouver les correspondances entre les instructions de ce programme et les instructions écrites sous scratch.
- Compléter le tableau ci-dessous :

Commande scratch	Commande Python
Stylo Aller à (-240,0) Relever le stylo Répéter 8 fois Stylo en position d'écriture avancer de pas à 50	stylo=pen stylo.goto(-240,0) stylo.up for i in range(8) stylo.dot stylo.forward vitessestylo = 50

2 / Ouvrir le logiciel edupython (ordinateur / Appli(V:) / Edupython / IDLE.exe)

- Recopier ce programme puis l'exécuter.
- Faire vérifier votre travail.
- Dans l'ENT, enregistrer votre fichier sous le nom act2_mouvement_rectiligne_uniforme_NOMS.py

Exécution du programme



3 / Mouvement rectiligne accéléré :

- Modifier le programme pour représenter un mouvement accéléré.
- Exécuter votre programme
- Dans l'ENT, enregistrer votre fichier sous le nom act2_mouvement_rectiligne_accélééré_NOMS.py

Copie d'écran :

```
• - from turtle import *
-
- #variable
• - stylo = Pen()
• - vitessestylo = 50 #en m/s
• - dt = 1 #en s
-
• - stylo.up()
• - stylo.goto(-240,0)
10
• - for i in range(8):
-     stylo.dot()
-     stylo.forward(vitessestylo*dt)
• -     vitessestylo = vitessestylo + 10
15
```

Exécution du programme

4 / Mouvement rectiligne ralenti :

- Modifier le programme pour représenter un mouvement ralenti.
- Exécuter votre programme.
- Dans l'ENT, enregistrer votre fichier sous le nom act2_mouvement_rectiligne_ralenti_NOMS.py

Copie d'écran :

```
from turtle import *

#variable
stylo = Pen()
vitessestylo = 50 #en m/s
dt = 1 #en s

stylo.up()
stylo.goto(-240,0)

for i in range(8):
    stylo.dot()
    stylo.forward(vitessestylo*dt)
    vitessestylo = vitessestylo-5
```

Exécution du programme

Activité 3

2 / Pour modifier la vitesse : ligne 11 : on remplace $v_0 = 5$ par $v_0 = 10$

3 / Mouvement accéléré

#boucle itérative

while i <= n:

$x_i = x_0 + v_0*dt$

$x_0 = x_i$

$v_i = v_0 + 5$ #il est possible d'augmenter la vitesse par simple incrémentation (ajout d'une valeur constante)

#ou encore en multipliant v_0 par un coefficient supérieur à 1. Selon le choix qui sera fait, il faudra ajuster les limites des axes et, certainement, l'échelle (scale) dans la commande quiver.

$v_0 = v_i$

 X.append(x_i)

 Y.append(y_0)

 valeur_vitesse.append(v_0)

 i += 1

4 / Mouvement ralenti :

#boucle itérative

while i <= n:

$x_i = x_0 + v_0*dt$

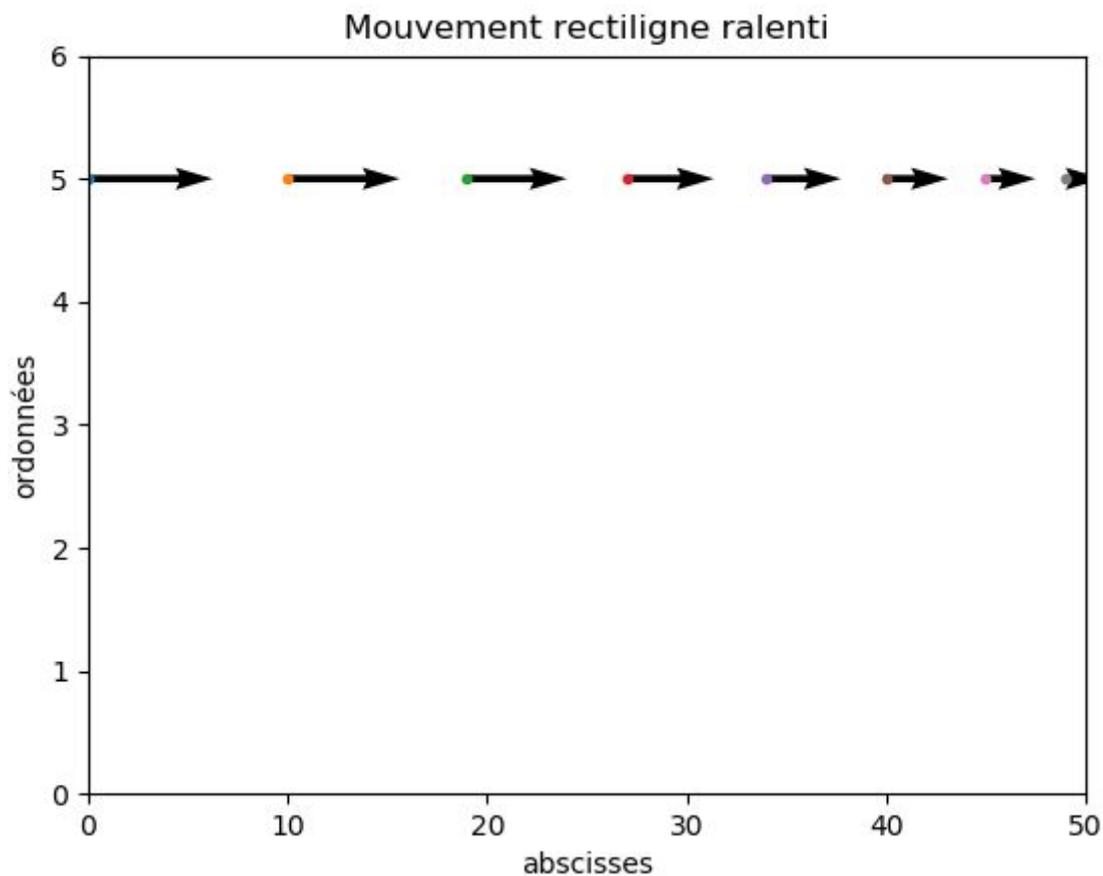
$x_0 = x_i$

$v_i = v_0 - 1$ #il est possible de diminuer la vitesse par simple incrémentation (on retranche une valeur

constante)

#ou encore en multipliant v_0 par un coefficient inférieur à 1. Selon le choix qui sera fait, il faudra ajuster les limites des axes et, certainement, l'échelle (scale) dans la commande quiver.

```
v_0 = v_i
X.append(x_i)
Y.append(y_0)
valeur_vitesse.append(v_0)
i += 1
```



Retour d'expérience :

Les plus-value pédagogiques (enseignants/élèves) :

Les freins :

Les leviers :

Les pistes pour aller plus loin ou généraliser la démarche :

Production d'élèves :

à venir