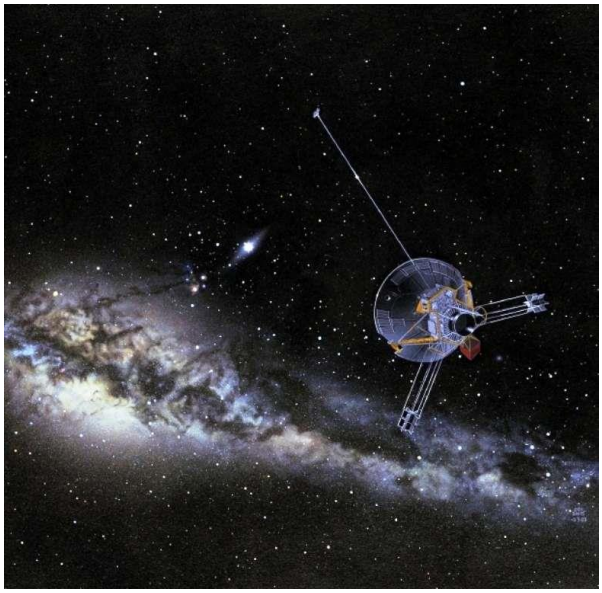
Activité n°3 – Scripter avec Blender : Trajectoire et principe d'inertie

Ouvrir le fichier **blender-meca-intro-inertie.blend** : cliquer ensuite sur la flèche de « lecture » de façon à lancer les calculs. Déplacer la Terre à l'aide des flèches de déplacement. Cette animation se déroule-t-elle dans l'espace ou sur Terre ?



Vue d'artiste d'une des sondes Pionner. © Nasa

En 1973, les sondes Pionner 10 et 11 sont lancées pour explorer les confins du système solaire. Les moteurs sont à l'arrêt depuis bien longtemps. En 1983, la sonde spatiale a franchi l'orbite de Neptune, la dernière planète du système solaire, et s'éloigne depuis en direction de l'étoile Aldébaran qu'elle atteindra dans 2 millions d'années.

Elle avance actuellement à la vitesse approximative de 12200 km.s^{-1} relativement au Soleil.

Comment avance-t-elle alors que ses moteurs ne fonctionnent plus ?

1 – Trajectoire

Qu'est-ce qu'un **mouvement** au sens usuel du terme ?

Une transition de positions d'un objet : il est dans une position A (x_A, y_A, z_A) au temps t_A et qui se trouve dans une position B (x_B, y_B, z_B) au temps t_B.

La trajectoire désigne alors l'ensemble des positions (x,y,z) prises par l'objet en mouvement dans le référentiel choisi.

Attention : nous avons vu dans l'activité précédente que l'allure de la trajectoire dépend du référentiel choisi. Il faut donc clairement le définir dès le début de l'étude si on veut pouvoir analyser objectivement les résultats d'une vidéo par exemple.

2 – Trajectoire rectiligne

Une trajectoire est dite **rectiligne** si l'allure de la trajectoire est une droite (ou un segment de droite).

01° Sur le document-réponse n°1, compléter le tableau et placer les positions d'un objet en mouvement ayant une vitesse constante de 1 cm par seconde. On ira jusqu'à une durée totale de 4s.

Le document que vous venez de créer ressemble alors à une **chronophotographie** : la superposition de plusieurs images d'un même objet durant son mouvement. Chaque image-photo est séparée du même intervalle de temps Δt : la durée entre chaque photo. Voici ci-dessous un exemple de chronophotographie :

Chronophotographie tirée de Wikipedia (Creative Commons - CC BY-SA 2.0 - Herve)

02° Lire les positions de l'objet présenté sur le document-réponse n°2. Remplir le tableau.

Il existe quatre grands types de mouvements rectilignes :

- Le **mouvement rectiligne uniforme** : la vitesse de l'objet est constante. Puisque la vitesse est constante, les différentes positions affichées sur une chronophotographie sont donc séparées par une distance constante.
- Le **mouvement rectiligne accéléré** : la vitesse de l'objet augmente au fil du temps. Les positions notées sur la chronophotographie sont donc de plus en plus éloignées les unes des autres.
- Le **mouvement rectiligne décéléré** : la vitesse de l'objet diminue au fil du temps. Les positions notées sur la chronophotographie sont donc de plus en plus proches les unes des autres.
- Le **mouvement rectiligne quelconque** : la vitesse augmente, diminue et/ou reste constante. Il existe différentes phases différentes.

03° A partir de l'observation des mouvements A, B, C et D, remplir le tableau réponse de la question 3.
Nous avons vu précédemment que la *distance Δd* parcourue, la *vitesse v* et la *durée Δt* du mouvement sont reliées par la formule suivante : $\Delta \boldsymbol{d} = \boldsymbol{v}.\Delta \mathbf{t}$
Si on recherche la vitesse, on peut utiliser la formule précédente et en déduire : $v = \frac{\Delta d}{\Delta t}$

04° Connaissant les unités de mesure du système international, en déduire l'unité SI de la vitesse v.
05° Déterminer la vitesse de déplacement v de l'objet du document-réponse n°1. Faire vérifier.

3 – Placer les objets avec Python dans Blender

Lors de l'activité précédente, nous avons créé une animation en utilisation manuellement l'interface de Blender. Cela vous a permis de comprendre que pour créer une animation, il fallait :

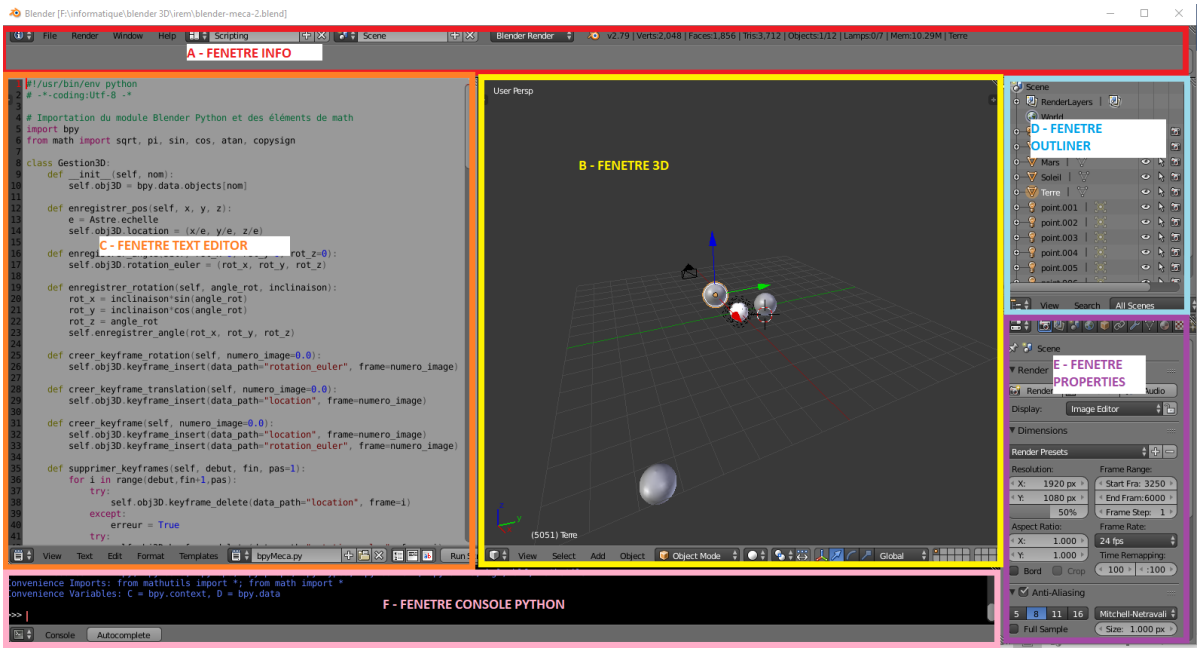
- Placer l'affichage sur la Frame voulue en utilisant le menu Timeline (en bas).
- Déplacer, déformer ou faire tourner l'objet voulu.
- Enregistrer la Keyframe en appuyant sur I.

Pour une petite animation, c'est gérable. S'il vaut réaliser 200 keyframes, cela serait trop long.
Nous allons donc aujourd'hui apprendre à scripter Blender en utilisant un langage de programmation : Python.

06° Ouvrir **blender-meca-3.blend**: il s'agit d'un fichier Blender dans lequel plusieurs objets ont déjà été créés :

Le Soleil – La Terre – La Lune – Mars
La caméra et 7 points d'émission de lumière nommés point.001 à point.007

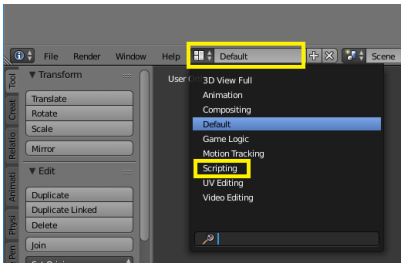
On obtient alors une fenêtre suivante, **légèrement différente** des activités précédentes :



L'écran est configuré pour permettre de visualiser 6 choses :

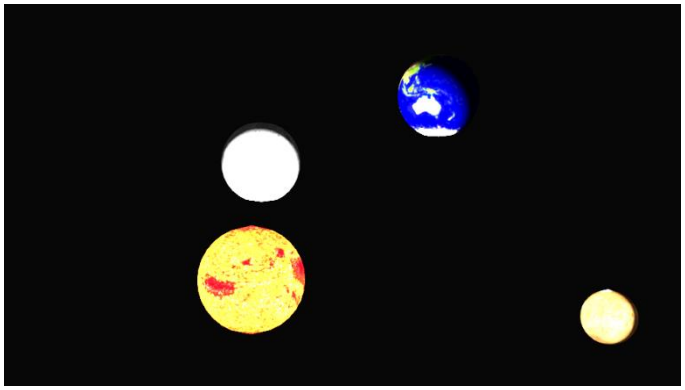
A – La fenêtre **Info** qui permet notamment d'enregistrer les fichiers 3D Blender (les fichiers .blend)
B – La fenêtre **3D** qui permet de visualiser le monde créé en 3D. Touche **T** pour la boite à outils à gauche et touche **N** pour l'affichage des propriétés géométriques à droite.
C – La fenêtre **Text Editor** qui va nous permettre d'écrire, d'ouvrir et de sauvegarder des scripts Python.
D – La fenêtre **Outliner** qui permet de sélectionner un objet 3D ou un outil (comme la caméra ou les éclairages)
E – La fenêtre **Properties** qui permet de modifier certains paramètres des objets (la puissance des lumières ect …)
F – La fenêtre **Console Python** qui permet d'exécuter des commandes Python en direct.

Vous pourriez revenir à la disposition des activités initiales en cliquant au bon endroit dans la fenêtre **INFO** :



Remarque : Vous pouvez agrandir ou rétrécir les fenêtres à l'aide de la souris en vous plaçant à l'intersection de deux fenêtres. Si vous réduisez trop une sous-fenêtre (comme la boite à outils), une petite icone + apparait en bordure de fenêtre. Elle vous permet d'afficher le menu qui a été réduit.

Si vous utilisez la touche F12 pour obtenir une image caméra, vous obtenez ceci pour l'instant :



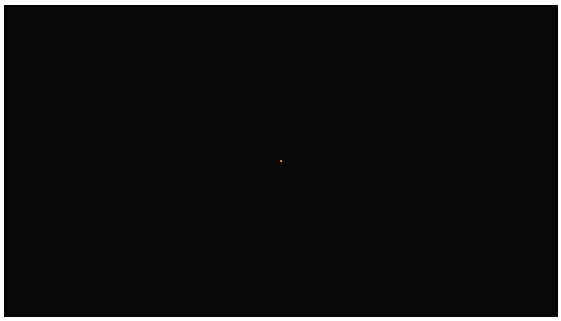
07° Dans **Text Editor**, sélectionner le fichier nommé **Script_act3_p2.py** si ce n'est pas le cas et lancer le script en cliquant sur le bouton **Run Script**. Vous aurez peut-être à agrandir la fenêtre **Text Editor**.

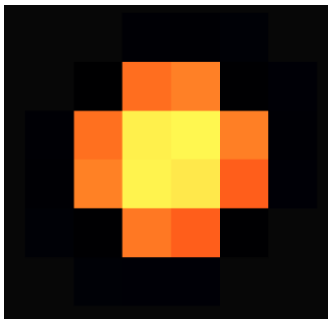
Vous devriez constater que les éléments 3D ont changé tout seul de taille et de position.

08° Lancer une simulation d'image avec F12.

Vous devriez obtenir la première image ci-dessous. On y voit le Soleil mais pas la Terre car à cette échelle, la Terre ne représente pas suffisamment de pixels. Les distances séparant les astres sont énormes et il est bien difficile d'afficher correctement tout ceci à la bonne échelle ET dans les bonnes proportions.

Si vous zoomez sur le Soleil, vous obtiendrez ceci :





09° Combien de pixels utilisent le logiciel pour représenter le Soleil à cette échelle ? La Terre est 100 fois plus petite environ. Pourquoi ne peut-on pas la représenter dans le logiciel à cette échelle ?

Vous pouvez enregistrer les images en utilisant le menu IMAGE qui possède une instruction SAVE. Pour revenir à la vue Text Editor, il faut placer la souris dans la fenêtre et appuyer sur ESC/ECHAP ou sélectionner à nouveau la bonne fenêtre (Text Editor) avec l'icône de fenêtre.

Que fait ce code ?

```
01  # 1 - Importation du module Blender Python et des éléments de math
02  from bpyMeca import Astre, Gestion
03  from math import cos, sin, pi
```

Le caractère # indique que la ligne est qu'un commentaire destiné au lecteur humain : l'ordinateur ne l'interprète absolument pas. Les commentaires sont en vert dans l'interpréteur Python de Blender.

Les lignes 2 et 3 permettent d'importer des classes et des fonctions qui ne sont pas automatiquement incluses de façon à ne pas avoir trop de choses en mémoire et gagner ainsi en rapidité d'exécution.

```
05  # 2 - Déclaration des grandeurs physiques
06  RAYON_SOLEIL = 695700E3 # rayon en m
07  RAYON_TERRE = 6371E3 # rayon en m
08  RAYON_LUNE = 1737E3 # rayon en m
09  RAYON_MARS = 3396E3 # rayon en m
10  ua = 150E9 # distance en m
11  DIST_TERRE_SOLEIL = ua # distance en m
12  DIST_TERRE_LUNE = 356700E3 # distance en m
13  DIST_MARS_SOLEIL = 249E9 # distance en m
```

On donne ici les valeurs de différentes grandeurs physiques qu'on stocke dans des **variables** : distance et rayon des astres. On pourra ensuite utiliser le contenu de la variable simplement en tapant son nom.

Attention, le tout est à exprimer en **mètres**.

10° Que contient la variable ***DIST_TERRE_SOLEIL*** ? Exprimer ce résultat en puissance de 10.

```
15  # 3 - Déclaration des constantes de simulation
16  CARREAU = ua/5 # On définit ici l'échelle d'un carreau BLENDER
17
18  # 4 - Création des objets Python associés aux objets 3D de Blender
19  Astre.definir_echelle(CARREAU)
```

Ici, on place (en ligne 16) dans la variable ***CARREAU*** l'échelle qu'on voudra donner à un carreau Blender.
En ligne 19, on utilise la méthode **definir_echelle** contenue dans la classe **Astre** pour rentrer l'échelle en mémoire.

11° En lisant le code, quelle est l'échelle d'un carreau Blender ?

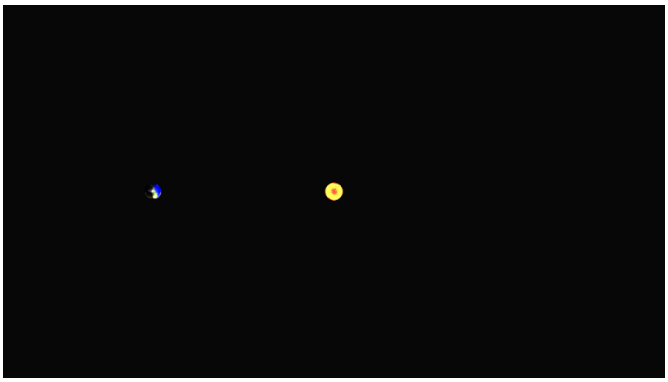
```
20  soleil = Astre("Soleil", zoom = 1, rayon = RAYON_SOLEIL)
21  terre = Astre("Terre", zoom = 1, rayon = RAYON_TERRE)
22  lune = Astre("Lune", zoom = 0, rayon = RAYON_LUNE)
23  mars = Astre("Mars", zoom = 0, rayon = RAYON_MARS)
24  camera = Gestion("Camera")
```

Ici, nous créons des objets Astre à partir des objets déjà créés dans Blender.

On donne leurs noms dans Blender, le zoom voulu (le facteur de grossissement par rapport à la réalité de l'échelle) et le rayon de l'astre.

De la même façon, on crée un objet Gestion qui permettra de gérer la caméra de Blender avec le script.

12° Puisqu'on ne voit quasiment pas le Soleil et la Terre à cette échelle, modifiez le code pour que la Terre apparaisse 1000 fois plus grosse et le Soleil apparaisse 10 fois plus gros que normalement. Relancer le script et faire une nouvelle 'photo' avec F12.



13° Comment indique-t-on via le script qu'on ne veut pas faire apparaitre Mars et la Lune ?

```
26  # 5 - Placement initial des objets
27  soleil.enregistrer_pos(x = 0, y = 0, z = 0)
28  terre.enregistrer_pos(x = DIST_TERRE_SOLEIL, y = 0, z = 0)
29  camera.enregistrer_pos(x = 0 , y= 0, z= DIST_TERRE_SOLEIL*4)
30  camera.viser(soleil)
```

Dans cette partie, nous allons placer les différents objets à l'aide d'un script plutôt que de le faire à la main.

On applique une méthode (rouge) sur un objet (orange) en plaçant un point entre les deux. Ainsi, la ligne 27 veut dire : appliquer la méthode **enregister_pos** sur l'objet **soleil**. Place l'objet aux coordonnées x,y,z données entre parenthèses.

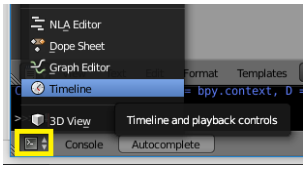
La méthode **viser** est un peu différente : elle s'applique sur **camera** et permet de l'orienter pour qu'elle fixe un astre en particulier. Ici, c'est **soleil**.

4 – Animer les objets avec Blender

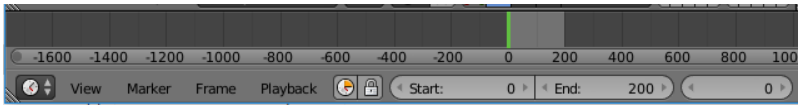
Pour animer, nous allons devoir déplacer la Terre et dire à Blender d'enregistrer les keyframes après avoir placer l'objet au bon endroit à l'aide de la méthode **enregister_pos**. Pour fixer la keyframe, il faudra utiliser la méthode **creer_keyframe** qui demande qu'on lui fournisse en argument le numéro de la frame à fixer. Exemple :

`terre.creer_keyframe(10)` va créer pour terre une keyframe sur la frame **10** en imposant la position de terre à la position où elle était à ce moment là dans le code.

14° Faire passer la fenêtre du bas de **Python Console** à **Timeline** en cliquant sur l'icône.



Vous allez alors voir apparaitre le menu qui permet de voir les points d'animation (**keyframes** dans Blender). Pour l'instant, vous n'en avez créé aucun et seule la raie verte de l'image actuelle apparait.



Créons une petite animation avec la méthode adaptée des instances d'**Astre** ou **Gestion** : la méthode **creer_keyframe**(numero= X) où X sera le numéro de la keyframe voulue.

15° Remplacer le code de la partie `# 5 - Placement initial des objets` par le code suivant :

```
# 5 - Placement initial des objets
a = DIST_TERRE_SOLEIL/5  # a est la distance constante parcourue entre 50 frames
terre.enregistrer_pos(x = DIST_TERRE_SOLEIL, y = 0, z = 0)
terre.creer_keyframe(0)

camera.enregistrer_pos(x = 0 , y= 0, z= DIST_TERRE_SOLEIL*4)
camera.viser(soleil)

terre.enregistrer_pos(x = DIST_TERRE_SOLEIL, y = 1*a, z = 0)
terre.creer_keyframe(50)

terre.enregistrer_pos(x = DIST_TERRE_SOLEIL, y = 2*a, z = 0)
terre.creer_keyframe(100)

terre.enregistrer_pos(x = DIST_TERRE_SOLEIL, y = 3*a, z = 0)
terre.creer_keyframe(150)

terre.enregistrer_pos(x = DIST_TERRE_SOLEIL, y = 4*a, z = 0)
terre.creer_keyframe(200)
```

16° Utiliser ce code pour visualiser l'animation obtenue.

Pour supprimer les keyframes (sans le faire à la main), on peut également le faire via Python :

17° Insérer la ligne de code suivante dans la partie `# 4 - Création des objets Python associés aux objets 3D de Blender` : nous supprimerons dans cette partie les keyframes créées par l'ancien script.

```
# 4 - Création des objets Python associés aux objets 3D de Blender
Astre.definir_echelle(CARREAU)
terre.supprimer_keyframes(0,500)
```

Cette simple ligne vous permet de supprimer les keyframes créées pour l'objet **terre** de la frame **0** à la frame **500**.

5 – Mouvement inertiel

Cette partie va vous permettre d'exploiter ce que vous avez vu précédemment. D'abord, un petit complément :

Principe d'inertie : si aucune force extérieure ne s'exerce sur le système, le mouvement est **rectiligne uniforme** et la vitesse est **constante**.

Activité 1 : (remarque – Les activités 1,2 et 3 sont à faire dans n'importe quel ordre).

18° La trajectoire de sphère symbolisant la Terre dans la simulation de la partie 4 est-elle caractéristique d'un système n'étant soumis à aucune force extérieure ?

19° A quoi ressemble la trajectoire réelle de la Terre ? Justifier alors que la Terre soit (ou non) soumise à une force extérieure.

20° Tracer sur le document-réponse le vecteur-force qui permettrait à la Terre d'avoir cette trajectoire. Remplir la case du référentiel d'étude en choisissant parmi héliocentrique, géocentrique et terrestre.

22° Donner alors une hypothèse sur les forces qui s'exercent encore ou non sur Pionner 10.

Activité 2 :

23° Utiliser la table à coussin autoporteur après avoir entendu les consignes de l'enseignant. Ces mobiles parviennent à compenser leurs poids. Lancer une acquisition. Justifier que son mouvement caractérise (ou non) un mouvement inertiel (c'est-à-dire un mouvement n'étant soumis à aucune force extérieure).

24° Quelle est la force qui est compensée sur ce mobile ?

25° Tracer sur le document-réponse les vecteurs-force que vous supposez exister. Remplir la case du référentiel d'étude en choisissant parmi héliocentrique, géocentrique et terrestre.

26° Expliquer alors pourquoi Pionner avance, elle, selon un mouvement inertiel.

Activité 3 :

Nous allons tenter de créer une animation montrant la trajectoire de la Terre si le Soleil cessait brusquement d'exister.

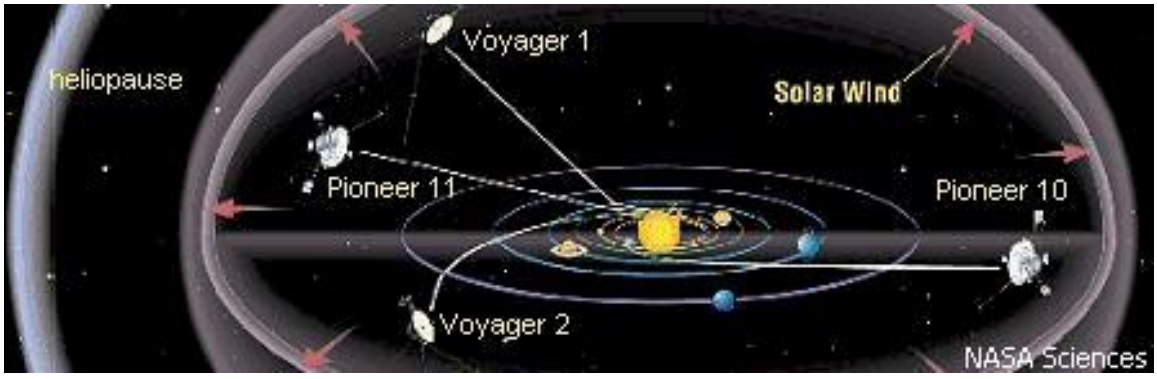
27° Comment calcule-t-on le périmètre p d'un cercle ? Connaissant la distance Terre-Soleil, déterminer alors la valeur du périmètre de la trajectoire terrestre. En combien de jours la Terre parvient-elle à faire le tour du Soleil ? En déduire la vitesse (supposée constante) de la Terre en mètre par jour puis en mètre par heure, puis en mètre par seconde.

28° Déterminer la distance que parcourt la Terre à cette vitesse en 1h.

On considère une animation de 24 images par seconde. On veut qu'une seconde affichée à l'écran corresponde à 1h de temps réel.

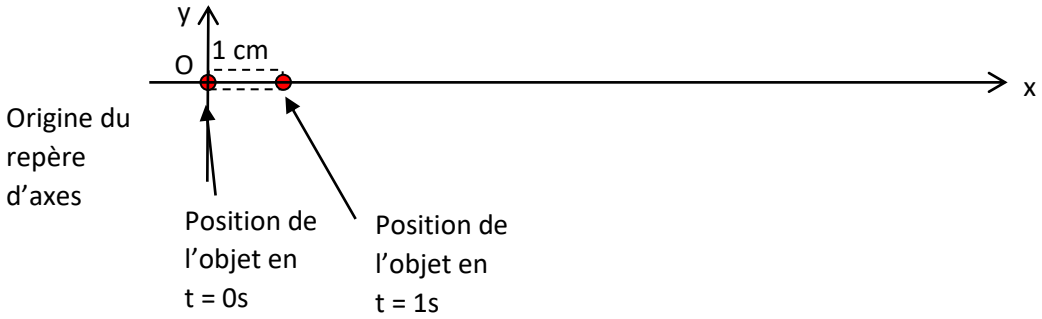
29° Réaliser l'animation montrant à quelle vitesse la Terre quitterait son orbite sans l'influence attractive du Soleil.

30° Au vu des trajectoires des différentes sondes envoyées par l'humanité, que peut-on dire de l'influence gravitationnelle du Soleil à grande distance ?



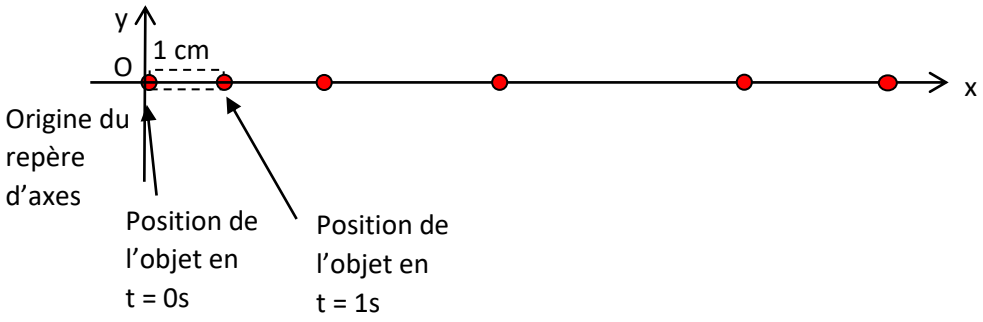
DOCUMENTS-REPONSE DE L'ACTIVITE N°3 DU CHAPITRE 5

Question 01 : Trajectoire rectiligne A ayant une vitesse constante de 1 cm par seconde selon x : v_X = 1 cm.s^{-1}.



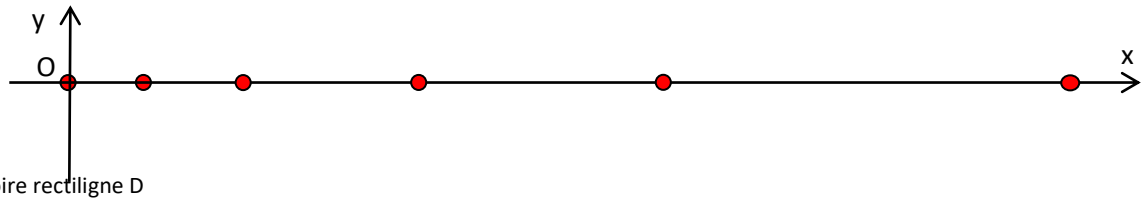
t(s)	0	1	2	3	4
x(cm)	0,0	1,0			
x(m)	0,000	0,010			

Question 02 : Trajectoire rectiligne B

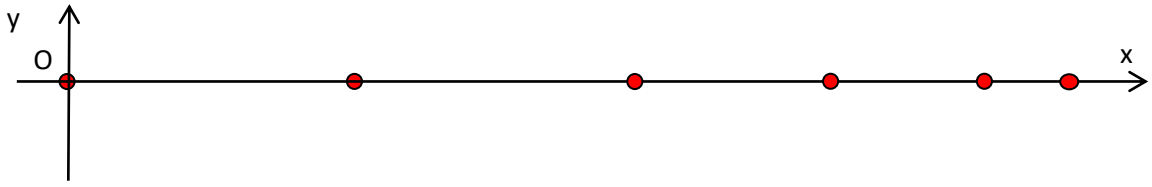


t(s)	0	1	2	3	4
x(cm)	0,0	1,0			
x(m)	0,000	0,010			

Question 03 : Trajectoire rectiligne C

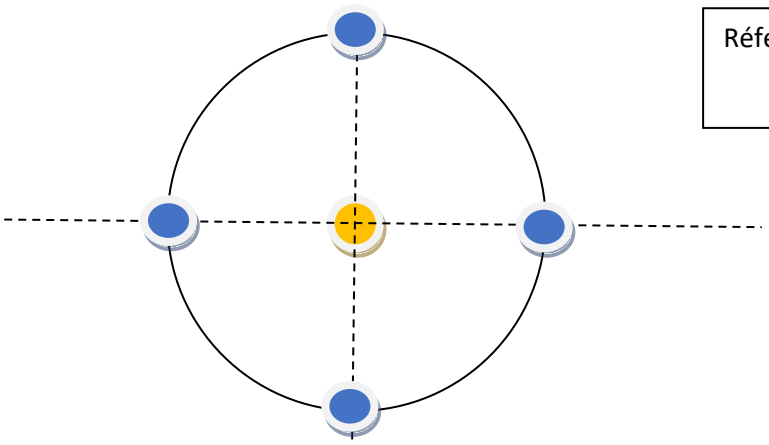


Question 03 : Trajectoire rectiligne D



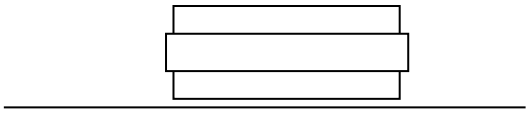
Trajectoire	A	B	C	D
Evolution de la vitesse				
Type de mouvement				

Question 20 :



Référentiel d'étude :

Question 25 :



Référentiel d'étude :