

Pesanteur

 Découverte

Voici un exercice clair, complet, prêt à l'emploi avec **Pygame** qui permettra à tes élèves de mieux comprendre la **chute libre** en simulant la pesanteur sur différents corps célestes.

Exercice : Simuler la chute libre sur différents corps célestes



Objectif de l'exercice

À la fin de cet exercice, tu seras capable :

- D'utiliser les formules de chute libre pour calculer la vitesse et l'altitude d'un objet en fonction du temps.
- De comprendre l'impact de la gravité différente selon les astres.
- De créer une simulation visuelle simple avec Pygame.



Théorie : formule de la chute libre

Lorsqu'un objet tombe en chute libre (sans frottements), sa vitesse et son altitude sont calculées ainsi :

- **Vitesse instantanée (v)** : $[v = g \cdot t]$
- **Altitude instantanée (y)** : $[y = h_0 - \frac{1}{2} \cdot g \cdot t^2]$

Avec :

- (g) = gravité (variable selon l'astre)
- (t) = temps écoulé (en secondes)
- (h_0) = hauteur initiale de l'objet (en mètres)

Théorie : formule de la chute libre

Lorsqu'un objet tombe en chute libre (sans frottements), sa vitesse et son altitude sont calculées ainsi :

- Vitesse instantanée (v) :

$$v = g \cdot t$$

- Altitude instantanée (y) :

$$y = h_0 - \frac{1}{2} \cdot g \cdot t^2$$

Corps céleste	Gravité (m/s ²)
Terre 🌍	9.81
Lune 🌕	1.62
Mars 🚀	3.72



Ton défi : complète la fonction

`calcul_chute`

Ci-dessous se trouve un **squelette complet** d'un programme Python avec Pygame.

Tu dois uniquement compléter la fonction `calcul_chute`.



Squelette de programme Python (avec Pygame)

Voici un code entièrement fonctionnel avec une place clairement indiquée pour ton calcul :

```
1  import pygame
2
3  # Paramètres initiaux
4  LARGEUR, HAUTEUR = 600, 600
5  FPS = 60
6
7  # Paramètres physiques
8  GRAVITE = 1.62 # Change cette valeur : 9.81 (Terre), 1.62 (Lune), 3.72 (Mars)
9  HAUTEUR_INITIALE = 500 # mètres
10
```

```

11 | pygame.init()
12 | fenetre = pygame.display.set_mode((LARGEUR, HAUTEUR))
13 | pygame.display.set_caption("Simulation de la chute libre")
14 | clock = pygame.time.Clock()
15 | font = pygame.font.SysFont("Arial", 18)
16 |
17 | def calcul_chute(gravite, hauteur_initiale, temps):
18 |     """
19 |     Complète cette fonction !
20 |     Retourne l'altitude actuelle et la vitesse actuelle de l'objet.
21 |     """
22 |
23 |     altitude = 0 # À compléter !
24 |     vitesse = 0 # À compléter !
25 |
26 |     return altitude, vitesse
27 |
28 | def main():
29 |     running = True
30 |     temps = 0
31 |     objet_en_chute = True
32 |
33 |     # Position graphique initiale de l'objet
34 |     x_objet = LARGEUR // 2
35 |     rayon_objet = 15
36 |
37 |     while running:
38 |         clock.tick(FPS)
39 |
40 |         for event in pygame.event.get():
41 |             if event.type == pygame.QUIT:
42 |                 running = False
43 |
44 |             if objet_en_chute:
45 |                 altitude, vitesse = calcul_chute(GRAVITE, HAUTEUR_INITIALE, temps)
46 |
47 |                 # Si l'objet touche le sol, on arrête la chute
48 |                 if altitude <= 0:
49 |                     altitude = 0
50 |                     objet_en_chute = False
51 |                 else:
52 |                     temps += 1 / FPS
53 |
54 |             # Conversion mètres vers pixels pour afficher à l'écran
55 |             pixels_par_metre = HAUTEUR / (HAUTEUR_INITIALE + 50)
56 |             y_objet = HAUTEUR - int(altitude * pixels_par_metre) - rayon_objet
57 |
58 |             # Affichage graphique
59 |             fenetre.fill((255, 255, 255)) # fond blanc
60 |
61 |             # Sol
62 |             pygame.draw.rect(fenetre, (100, 200, 100), (0, HAUTEUR - 10, LARGEUR, 10))
63 |
64 |             # Objet
65 |             pygame.draw.circle(fenetre, (255, 0, 0), (x_objet, y_objet), rayon_objet)
66 |
67 |             # Affichage des infos en haut à droite

```

```

65     infos = [
66         f"Altitude : {altitude:.2f} m",
67         f"Vitesse : {vitesse:.2f} m/s",
68         f"Gravité : {GRAVITE:.2f} m/s²"
69     ]
70
71     for i, texte in enumerate(infos):
72         surface_texte = font.render(texte, True, (0, 0, 0))
73         largeur_texte, hauteur_texte = surface_texte.get_size()
74         fenetre.blit(surface_texte, (LARGEUR - largeur_texte - 10, 10 + i * (hauteur_texte
75
76     pygame.display.flip()
77
78     pygame.quit()
79
80 if __name__ == "__main__":
81     main()

```



Synthèse : ce qu'il faut retenir

- En chute libre, l'objet accélère constamment sous l'effet de la gravité.
- L'altitude diminue quadratiquement en fonction du temps.
- La vitesse augmente linéairement avec le temps.

Ce programme visuel et interactif aide à comprendre clairement comment la gravité influence la chute libre sur différents corps célestes.

À toi de jouer ! 🎮