

Trajectoires

 Découverte

Voici un exercice prêt à l'emploi destiné aux élèves du secondaire math-sciences, permettant de coder en Python un calculateur simple de trajectoire d'un projectile. Cet exercice met en pratique les compétences en physique, mathématiques et programmation.

Exercice : Calcul de la trajectoire d'un projectile avec Python et Pygame

Objectif

À la fin de cet exercice, tu seras capable :

- D'utiliser des formules de physique pour calculer la trajectoire d'un projectile.
- D'intégrer une fonction mathématique dans un jeu codé en Python.
- De visualiser le résultat de ton calcul grâce à la bibliothèque Pygame.

Rappels théoriques

Formules essentielles à connaître

Pour calculer la trajectoire d'un projectile (ex. : un boulet de canon), on utilise les formules suivantes issues de la physique du mouvement :

- **Décomposition de la vitesse initiale :**

Quand tu lances un projectile à une vitesse initiale (v), avec un angle de lancement (α) :

$$[v_x = v \cdot \cos(\alpha) \quad ; \quad v_y = v \cdot \sin(\alpha)]$$

- **Position en fonction du temps :**

La position du projectile à l'instant (t) avec une gravité (g) est donnée par :

$$[x(t) = v_x \cdot t]$$

$$[y(t) = v_y \cdot t - \frac{1}{2} g \cdot t^2]$$

Où :

- (v) est la vitesse initiale (en m/s),
- (α) l'angle (en radians),
- (g) la gravité (généralement (9.81, m/s²)),
- (t) le temps écoulé depuis le lancement (en secondes).

• Étape 2 :

On applique les formules suivantes pour décomposer la vitesse initiale :

$$v_x = v \cdot \cos(\alpha)$$

$$v_y = v \cdot \sin(\alpha)$$

• Étape 3 :

Le mouvement horizontal du projectile n'est pas affecté par la gravité (en absence de résistance de l'air). La formule utilisée est donc :

$$x = v_x \cdot t$$

• Étape 4 :

Le mouvement vertical est affecté par la gravité (le projectile est constamment attiré vers le bas). On utilise donc la formule :

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

Ton défi : Complète la fonction de calcul

Voici le squelette du programme de jeu. À toi de coder la fonction `calcul_trajectoire` en utilisant les formules ci-dessus !

Squelette du programme Python (avec Pygame)

```
1  import pygame
2  import math
3
4  # Paramètres du jeu
5  LARGEUR, HAUTEUR = 1900, 1000
6  FPS = 60
7  GRAVITE = 9.81
8
9  pygame.init()
```

```

10 fenetre = pygame.display.set_mode((LARGEUR, HAUTEUR))
11 pygame.display.set_caption("Simulation de trajectoire")
12 clock = pygame.time.Clock()
13
14
15 projectile_en_vol = True
16
17 def calcul_trajectoire(angle, vitesse, gravite, temps):
18     """
19     Calcule la position (x,y) du projectile à l'instant 'temps'.
20     Angle en degrés, vitesse en m/s, gravité en m/s², temps en s.
21     Retourne un tuple (x, y).
22     """
23
24     # IMPLÉMENTE LE CODE
25
26     return x, y
27
28 # Boucle principale du jeu
29 def main():
30     running = True
31     temps = 0
32     angle = 70 # Angle de lancement en degrés
33     vitesse = 150 # Vitesse initiale en m/s
34
35     trajectoire = [] # Liste des points calculés
36     projectile_en_vol = True
37
38     # Configuration de la police pour afficher les informations
39     font = pygame.font.SysFont("Arial", 18)
40
41     while running:
42         clock.tick(FPS)
43
44         for event in pygame.event.get():
45             if event.type == pygame.QUIT:
46                 running = False
47
48         # Calcule la trajectoire point par point toutes les 0.1 secondes
49         if projectile_en_vol:
50             x, y = calcul_trajectoire(angle, vitesse, GRAVITE, temps)
51
52             # Conversion coordonnées réelles vers coordonnées pygame (inverse y)
53             if y >= 0:
54                 pygame_x = int(x)
55                 pygame_y = int(HAUTEUR - y)
56                 trajectoire.append((pygame_x, pygame_y))
57                 temps += 0.05
58             else:
59                 projectile_en_vol = False
60
61         # Affichage
62         fenetre.fill((255, 255, 255)) # fond blanc
63
64         # Dessiner la trajectoire
65         for point in trajectoire:
66             pygame.draw.circle(fenetre, (255, 0, 0), point, 3)

```

```

64         # Afficher vitesse, altitude, temps en haut à droite
65         infos = [
66             f"Vitesse initiale : {vitesse:.1f} m/s",
67             f"Altitude : {max(0, y):.1f} m",
68             f"Temps écoulé : {temps:.1f} s"
69         ]
70
71         # Affichage des infos avec un léger espacement vertical
72         for i, texte in enumerate(infos):
73             surface_texte = font.render(texte, True, (0, 0, 0))
74             largeur_texte, hauteur_texte = surface_texte.get_size()
75             fenetre.blit(surface_texte, (LARGEUR - largeur_texte - 10, 10 + i * (hauteur_texte
76
77         pygame.display.flip()
78
79     pygame.quit()
80
81 if __name__ == "__main__":
82     main()
83
84

```

Instructions détaillées pour la fonction

Pour compléter `calcul_trajetoire`, tu dois :

1. Convertir l'angle donné (en degrés) en radians à l'aide de `math.radians(angle)`.
2. Décomposer la vitesse en composantes horizontale (`v_x`) et verticale (`v_y`).
3. Calculer les coordonnées `x` et `y` à l'aide des formules données.
4. Retourner `(x, y)`.

Astuce : Teste d'abord ta fonction avec des valeurs simples pour vérifier qu'elle fonctionne correctement.

Exemple de résultat attendu

Si tu utilises les valeurs :

- `angle = 45°`
- `vitesse = 50 m/s`
- `gravité = 9.81 m/s2`

Tu devrais observer à l'écran une courbe représentant la trajectoire du projectile (en forme de parabole).

Synthèse : ce qu'il faut retenir

- La trajectoire d'un projectile est déterminée par deux composantes distinctes de mouvement (horizontal et vertical).
- Les positions sont calculées en utilisant des fonctions trigonométriques (`sin` , `cos`) et des équations de physique.
- Le temps permet d'animer le déplacement en calculant successivement les positions du projectile.
- Toujours valider tes résultats intermédiaires pour éviter les erreurs.

À toi de jouer ! Bonne programmation !  