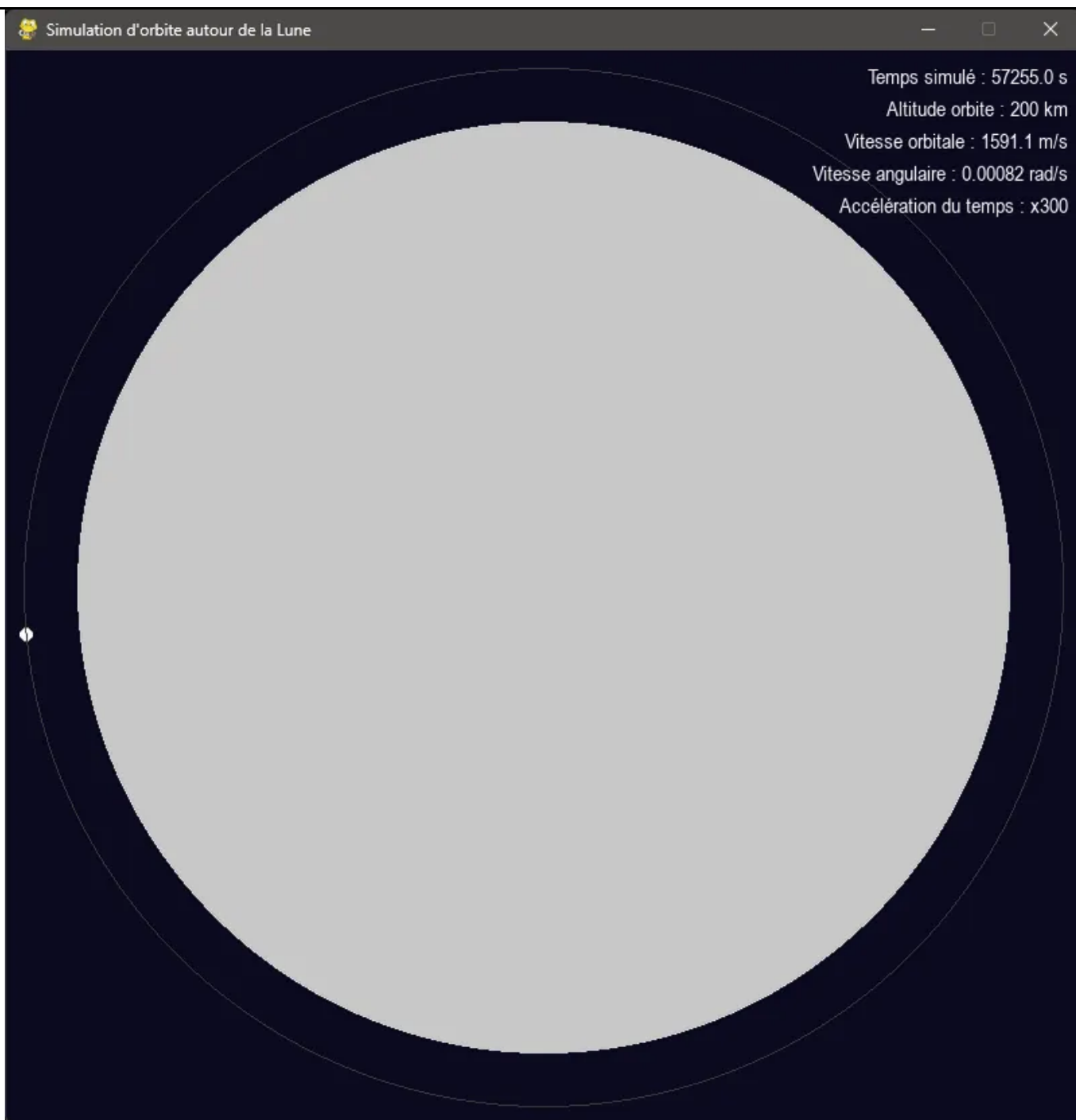


Orbite

 Découverte

Principe :

Un satellite en orbite autour d'une planète suit un mouvement circulaire uniforme décrit par la loi de la gravitation universelle.

Formules :

- Vitesse orbitale :

$$v = \sqrt{\frac{GM}{R}}$$

- Position (en coordonnées polaires) :

$$x(t) = R \cdot \cos(\omega t), \quad y(t) = R \cdot \sin(\omega t)$$

avec :

- G : constante gravitationnelle ($6.674 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$)
- M : masse de la planète (kg)
- R : rayon orbital (mètres)
- ω : vitesse angulaire $\frac{v}{R}$

```
1  import pygame
2  import math
3
4  # Paramètres graphiques
5  LARGEUR, HAUTEUR = 800, 800
6  FPS = 60
7
8  # Paramètres physiques (Lune)
9  G = 6.674e-11          # Constante universelle (toujours la même)
10 M = ...                 # Masse de l'astre central (kg)
11 R = ...                 # Rayon de l'astre central (m)
12 altitude = ...         # Altitude de l'objet en orbite (m)
13 rayon_orbite = R + altitude # Rayon orbital total (m)
14
15
16 # Échelle de conversion (pixels/mètre)
17 ECHELLE = 1 / 5000
18
19 pygame.init()
20 fenetre = pygame.display.set_mode((LARGEUR, HAUTEUR))
21 pygame.display.set_caption("Simulation d'orbite autour de la Lune")
22 clock = pygame.time.Clock()
23 font = pygame.font.SysFont("Arial", 16)
24
25 def calcul_orbite(G, M, R, temps):
26     # Calcule vitesse orbitale et position
27
```

```

28     # Ecris tes calculs ici
29
30     return x, y, v, omega
31
32 def main():
33     running = True
34     temps = 0
35     facteur_acceleration = 300 # Pour visualiser l'orbite plus vite
36
37     centre_x, centre_y = LARGEUR // 2, HAUTEUR // 2
38
39     while running:
40         clock.tick(FPS)
41         for event in pygame.event.get():
42             if event.type == pygame.QUIT:
43                 running = False
44
45         temps += (1/FPS) * facteur_acceleration
46
47         x, y, vitesse_orbitale, omega = calcul_orbite(G, M, R, temps)
48
49         satellite_x = centre_x + int(x * ECHELLE)
50         satellite_y = centre_y + int(y * ECHELLE)
51
52         # Affichage
53         fenetre.fill((10, 10, 30)) # Espace sombre
54         pygame.draw.circle(fenetre, (200, 200, 200), (centre_x, centre_y), int(RAYON_LUNE * ECHELLE))
55         pygame.draw.circle(fenetre, (255, 255, 255), (satellite_x, satellite_y), 5) # Satellite
56
57         # Orbite tracée
58         pygame.draw.circle(fenetre, (60, 60, 60), (centre_x, centre_y), int(RAYON_ORBITE * ECHELLE))
59
60         # Affichage des infos
61         infos = [
62             f"Temps simulé : {temps:.1f} s",
63             f"Altitude orbite : {ALTITUDE/1000:.0f} km",
64             f"Vitesse orbitale : {vitesse_orbitale:.1f} m/s",
65             f"Vitesse angulaire : {omega:.5f} rad/s",
66             f"Accélération du temps : x{facteur_acceleration}"
67         ]
68
69         for i, texte in enumerate(infos):
70             surface_texte = font.render(texte, True, (255, 255, 255))
71             largeur_texte, hauteur_texte = surface_texte.get_size()
72             fenetre.blit(surface_texte, (LARGEUR - largeur_texte - 10, 10 + i*(hauteur_texte + 10)))
73
74         pygame.display.flip()
75
76     pygame.quit()
77
78 if __name__ == "__main__":
79     main()
80

```

Paramètres:

Lune autours de la terre

```
1 | M = 5.972e24          # Masse Terre (kg)
2 | R = 6.371e6           # Rayon Terre (m)
3 | altitude = 3.844e8 - R # Distance Terre-Lune = 384400 km
4 | rayon_orbite = R + altitude
```

Période orbitale attendue : $\approx 27,3$ jours (réaliste)

Satellite Starlink autour de la Terre (~550 km d'altitude)

```
1 | M = 5.972e24          # Masse Terre (kg)
2 | R = 6.371e6           # Rayon Terre (m)
3 | altitude = 5.5e5      # Altitude Starlink = 550 km
4 | rayon_orbite = R + altitude
```

Satellite lunaire bas (autour de la Lune, à 100 km)

```
1 | M = 7.3477e22         # Masse Lune (kg)
2 | R = 1.737e6           # Rayon Lune (m)
3 | altitude = 1.0e5      # Altitude orbite = 100 km
4 | rayon_orbite = R + altitude
```

Terre autour du Soleil

```
1 | M = 1.989e30          # Masse Soleil (kg)
2 | R = 6.9634e8          # Rayon Soleil (m) - valeur indicative uniquement
3 | altitude = 1.496e11 - R # Distance moyenne Soleil-Terre = 149.6 millions km
4 | rayon_orbite = R + altitude
```