

# Robot

 Découverte

Voici un **code de base complet en Python avec `pygame`** qui simule un petit **robot équipé de capteurs virtuels** :

- Vitesse (en fonction du déplacement)
- Température (aléatoire dans une plage)
- Capteur de distance (ultrason simulé par un rayon)
- Capteur infrarouge (détection de "sources de chaleur")



## Objectif pédagogique

Comprendre comment un robot peut lire et interpréter des capteurs dans son environnement.



## Dépendances

Installe Pygame si nécessaire :

```
1 | pip install pygame
```



## Code Python complet

```
1 | import pygame
2 | import random
3 | import math
4 |
5 | # Initialisation
6 | pygame.init()
7 | WIDTH, HEIGHT = 800, 600
8 | screen = pygame.display.set_mode((WIDTH, HEIGHT))
9 | pygame.display.set_caption("Simulation robot avec capteurs")
10 | clock = pygame.time.Clock()
```

```

11
12 # Robot
13 robot = pygame.Rect(100, 100, 40, 40)
14 robot_speed = 2
15 robot_angle = 0 # direction en degrés
16
17 # Obstacles
18 obstacles = [
19     pygame.Rect(300, 200, 100, 50),
20     pygame.Rect(600, 400, 50, 150)
21 ]
22
23 # Sources de chaleur (détectables par IR)
24 heat_sources = [
25     pygame.Rect(700, 100, 20, 20),
26     pygame.Rect(150, 500, 20, 20)
27 ]
28
29 font = pygame.font.SysFont(None, 24)
30
31 def get_temperature():
32     return round(random.uniform(18, 28), 1)
33
34 def get_velocity():
35     return robot_speed # constant ici, mais peut varier
36
37 def get_distance_to_obstacle():
38     """Renvoie la distance jusqu'au 1er obstacle en face"""
39     ray_length = 150
40     for i in range(ray_length):
41         x = robot.centerx + i * math.cos(math.radians(robot_angle))
42         y = robot.centery + i * math.sin(math.radians(robot_angle))
43         point = pygame.Rect(x, y, 2, 2)
44         if any(point.colliderect(obs) for obs in obstacles):
45             return i
46     return ray_length
47
48 def detect_infrared():
49     """Détection d'une source de chaleur proche"""
50     for heat in heat_sources:
51         dist = math.hypot(robot.centerx - heat.centerx, robot.centery - heat.centery)
52         if dist < 100:
53             return True
54     return False
55
56 def draw_environment():
57     screen.fill((30, 30, 30))
58
59     # Obstacles
60     for obs in obstacles:
61         pygame.draw.rect(screen, (150, 150, 150), obs)
62
63     # Sources de chaleur
64     for heat in heat_sources:
65         pygame.draw.rect(screen, (255, 0, 0), heat)
66

```

```

67     # Robot
68     pygame.draw.rect(screen, (0, 200, 255), robot)
69
70     # Capteur distance : rayon
71     dist = get_distance_to_obstacle()
72     end_x = robot.centerx + dist * math.cos(math.radians(robot_angle))
73     end_y = robot.centery + dist * math.sin(math.radians(robot_angle))
74     pygame.draw.line(screen, (0, 255, 0), robot.center, (end_x, end_y), 2)
75
76     # Données capteurs
77     temp = get_temperature()
78     vel = get_velocity()
79     ir = detect_infrared()
80
81     capteurs = [
82         f"🌡 Température : {temp} °C",
83         f"🚀 Vitesse : {vel} px/s",
84         f"📏 Distance obstacle : {dist} px",
85         f"🔥 Infrarouge : {'Oui' if ir else 'Non'}"
86     ]
87
88     for i, txt in enumerate(capteurs):
89         img = font.render(txt, True, (255, 255, 255))
90         screen.blit(img, (10, 10 + i * 20))
91
92     running = True
93     while running:
94         clock.tick(60)
95         keys = pygame.key.get_pressed()
96
97         # Déplacement
98         if keys[pygame.K_LEFT]:
99             robot_angle -= 3
100         if keys[pygame.K_RIGHT]:
101             robot_angle += 3
102         if keys[pygame.K_UP]:
103             robot.x += robot_speed * math.cos(math.radians(robot_angle))
104             robot.y += robot_speed * math.sin(math.radians(robot_angle))
105
106         for event in pygame.event.get():
107             if event.type == pygame.QUIT:
108                 running = False
109
110         draw_environment()
111         pygame.display.flip()
112
113     pygame.quit()

```



## Explications

| Élément                                 | Description                                                      |
|-----------------------------------------|------------------------------------------------------------------|
| <code>robot_angle</code>                | Contrôle l'orientation du robot                                  |
| <code>get_temperature()</code>          | Simule une température entre 18 et 28°C                          |
| <code>get_velocity()</code>             | Renvoie une vitesse constante                                    |
| <code>get_distance_to_obstacle()</code> | Simule un capteur ultrason ou LIDAR avec un rayon jusqu'à 150 px |
| <code>detect_infrared()</code>          | Détecte une source de chaleur à moins de 100 px                  |
| <code>pygame.draw.line()</code>         | Affiche visuellement le rayon du capteur de distance             |