

```

import cv2
import mediapipe as mp
import numpy as np

# Initialisation de MediaPipe et de la détection des poses
mp_drawing = mp.solutions.drawing_utils
mp_pose = mp.solutions.pose

# Fonction pour calculer l'angle entre trois points (p0, p1, p2)
def calculate_angle(p0, p1, p2):
    v1 = np.array(p0) - np.array(p1)
    v2 = np.array(p2) - np.array(p1)
    cosine_angle = np.dot(v1, v2) / (np.linalg.norm(v1) *
np.linalg.norm(v2))
    angle = np.degrees(np.arccos(cosine_angle))
    return angle

# Fonction pour convertir les coordonnées de Mediapipe en coordonnées
pour Pepper
def convert_coordinates(landmark, image_shape):
    # Normalisation des coordonnées
    x = landmark.x * image_shape[1]
    y = landmark.y * image_shape[0]
    z = landmark.z
    # Conversion des coordonnées selon le système de Pepper
    # (Cela peut être ajusté selon le référentiel de Pepper)
    x_pepper = x / image_shape[1]
    y_pepper = y / image_shape[0]
    z_pepper = z
    return x_pepper, y_pepper, z_pepper

# Accès à la caméra de l'ordinateur
cap = cv2.VideoCapture(0)

with mp_pose.Pose(min_detection_confidence=0.5,
min_tracking_confidence=0.5) as pose:
    while cap.isOpened():
        ret, frame = cap.read()

        if not ret:
            break

        # Conversion de l'image en RGB pour MediaPipe
        image_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image_rgb.flags.writeable = False

        # Processus de détection des poses
        results = pose.process(image_rgb)

        # Récupération des dimensions de l'image

```

```

image_shape = frame.shape

# Si des landmarks sont détectés
if results.pose_landmarks:
    landmarks = results.pose_landmarks.landmark

    # Récupérer les points pour le coude gauche (landmark 13),
    # épaule gauche (11) et poignet gauche (15)
    left_shoulder = [landmarks[11].x, landmarks[11].y,
landmarks[11].z]
    left_elbow = [landmarks[13].x, landmarks[13].y,
landmarks[13].z]
    left_wrist = [landmarks[15].x, landmarks[15].y,
landmarks[15].z]

    # Calculer l'angle Roll au niveau du coude gauche
    roll_angle_left_elbow = calculate_angle(left_shoulder,
left_elbow, left_wrist)
    print(f"Angle de Roll du coude gauche:
{roll_angle_left_elbow}")

    # Conversion des coordonnées de Mediapipe en coordonnées
    # pour Pepper
    left_elbow_pepper = convert_coordinates(landmarks[13],
image_shape)
    print(f"Coordonnées du coude gauche pour Pepper :
{left_elbow_pepper}")

    # Affichage des landmarks sur l'image
    mp_drawing.draw_landmarks(frame, results.pose_landmarks,
mp_pose.POSE_CONNECTIONS)

    # Affichage du flux vidéo
    cv2.imshow('Mediapipe Feed', frame)

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

```