import cv2  
import numpy as np  
import HandTrackingModule as htm  
import time  
import autopy  
  
  
###############################  
wCam, hCam = 640, 480  
frameR = 100 # Frame Reduction  
smoothening = 5  
###############################  
  
pTime = 0  
plocX, plocY = 0, 0  
clocX, clocY = 0, 0  
  
  
cap = cv2.VideoCapture(0)  
cap.set(3, wCam)  
cap.set(4, hCam)  
detector = htm.handDetector(maxHands=1)  
#print(wScr, hScr)  
while True:  
 # 1. Find hand landmarks  
 wScr, hScr = autopy.screen.size()  
 success, img = cap.read()  
 img = detector.findHands(img)  
 lmList, bbox = detector.findPosition(img)  
  
 # 2. Get the tip of index and middle fingers  
 if len(lmList)!=0:  
 x1, y1 = lmList[8][1:]  
 x2, y2 = lmList[8][1:]  
 #print(x1,y1,x2,y2)  
  
  
 # 3. Check which fingers are up  
 fingers = detector.fingersUp()  
 #print(fingers)  
 cv2.rectangle(img, (frameR, frameR), (wCam - frameR, hCam - frameR), (255, 0, 255), 2)  
  
 # 4. Only index finger: Moving mode  
 if fingers[1] == 1 and fingers[2] == 0:  
 # 5. Convert Coordinates  
 # cv2.rectangle(img, (frameR, frameR), (wCam-frameR, hCam-frameR), (255, 0, 255), 2)  
 x3 = np.interp(x1, (frameR, wCam-frameR), (0, wScr))  
 y3 = np.interp(y1, (frameR, hCam-frameR), (0, hScr))  
  
  
 # 6. Smoothen the values  
 clocX = plocX + (x3 - plocX) / smoothening  
 clocY = plocY + (y3 - plocY) / smoothening  
 # 7. Move mouse  
 autopy.mouse.move(wScr-clocX, clocY)  
 cv2.circle(img, (x1, y1), 15, (255, 0, 255), cv2.FILLED)  
 plocX, plocY = clocX, clocY  
 # 8. Both index and middle fingers are up: Clicking mode  
 if fingers[1] == 1 and fingers[2] == 1:  
 # 9. Find distance between fingers  
 length, img, lineInfo = detector.findDistance(8, 12, img)  
 print(length)  
 # 10. Click mouse if distance is short  
 if length < 40:  
 cv2.circle(img, (lineInfo[4], lineInfo[5]), 15, (0, 255, 0), cv2.FILLED)  
 autopy.mouse.click()  
  
  
 # 10. Click mouse if distance is short  
 # 11. Frame rate  
 cTime = time.time()  
 fps = 1 / (cTime - pTime)  
 pTime = cTime  
 cv2.putText(img, str(int(fps)), (20, 50), cv2.FONT\_HERSHEY\_PLAIN, 3, (255, 0, 0), 3)  
  
 # 12. Display  
 cv2.imshow("Image", img)  
 cv2.waitKey(1)