```
import cv2
import mediapipe as mp
import pyautogui

def smooth(coordinates, alpha=0.5):
    if not hasattr(smooth, 'previous_coordinates'):
        smooth.previous_coordinates = coordinates

    smoothed_coordinates = []
    for (x, y), (prev_x, prev_y) in zip(coordinates,
smooth.previous_coordinates):
        smoothed_x = int(prev_x * alpha + x * (1 - alpha))
        smoothed_y = int(prev_y * alpha + y * (1 - alpha))
        smoothed_coordinates.append((smoothed_x, smoothed_y))

    smooth.previous_coordinates = smoothed_coordinates
    return smoothed_coordinates

cam = cv2.VideoCapture(0)
face_mesh = mp.solutions.face_mesh.FaceMesh(refine_landmarks=True)
screen_w, screen_h = pyautogui.size()

while True:
    _, frame = cam.read()
    frame = cv2.flip(frame, 1)
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    output = face_mesh.process(rgb_frame)
    landmark_points = output.multi_face_landmarks

    frame_h, frame_w, _ = frame.shape

    if landmark_points:
        landmarks = landmark_points[0].landmark

        # Update the range to 468, which is the correct number of
landmarks
        coordinates = [(int(landmark.x * frame_w), int(landmark.y *
frame_h)) for landmark in landmarks[:468]]
        smoothed_coordinates = smooth(coordinates)

        for (x, y) in smoothed_coordinates:
            cv2.circle(frame, (x, y), 3, (0, 255, 0))

        x, y = smoothed_coordinates[1]  # Index 1 is the control point

        # Adjust the click threshold as needed
        click_threshold = 5

        if abs(coordinates[145][1] - coordinates[159][1]) <
click_threshold:
            pyautogui.click()
            pyautogui.sleep(0.2)  # Add a small delay for the click
action
```

```
        screen_x = int(screen_w * x / frame_w)
        screen_y = int(screen_h * y / frame_h)
        pyautogui.moveTo(screen_x, screen_y)

    cv2.imshow('Eye Controlled Mouse', frame)

    # Exit the loop when 'q' key is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the camera and destroy OpenCV windows
cam.release()
cv2.destroyAllWindows()
```