



Nombre: Edgar Javier Fregoso Cuarenta

Registro: 22310285

Materia : visión Artificial

Proyecto

```
import cv2

import mediapipe as mp

import random

import time


# Inicializa MediaPipe

mp_hands = mp.solutions.hands

mp_drawing = mp.solutions.drawing_utils

hands = mp_hands.Hands(min_detection_confidence=0.7,
min_tracking_confidence=0.7)


# Cámara

cap = cv2.VideoCapture(0)


# Mapeo de gestos

def detectar_jugada(landmarks):

    dedos = []


    # Dedos índice a meñique

    for i in [8, 12, 16, 20]:

        if landmarks.landmark[i].y < landmarks.landmark[i - 2].y:

            dedos.append(1)

        else:

            dedos.append(0)


    # Pulgar
```

```
if landmarks.landmark[4].x > landmarks.landmark[3].x:
    dedos.insert(0, 1)
else:
    dedos.insert(0, 0)

if sum(dedos) == 0:
    return "Piedra"
elif sum(dedos) == 5:
    return "Papel"
elif dedos[1] == 1 and dedos[2] == 1 and sum(dedos) == 2:
    return "Tijera"
else:
    return "Gesto desconocido"
```

# Función para determinar el ganador

```
def determinar_ganador(jugador, computadora):
    if jugador == computadora:
        return "Empate"
    elif (jugador == "Piedra" and computadora == "Tijera") or \
        (jugador == "Papel" and computadora == "Piedra") or \
        (jugador == "Tijera" and computadora == "Papel"):
        return "¡Ganaste!"
    else:
        return "Perdiste"
```

# Lista de opciones

```
opciones = ["Piedra", "Papel", "Tijera"]
```

```
# Temporizador
```

```
contador = 5
```

```
jugando = False
```

```
jugada_jugador = None
```

```
resultado = ""
```

```
tiempo_inicio = None
```

```
while cap.isOpened():
```

```
    ret, frame = cap.read()
```

```
    if not ret:
```

```
        break
```

```
    frame = cv2.flip(frame, 1)
```

```
    imagen_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

```
    resultado_manos = hands.process(imagen_rgb)
```

```
    if resultado_manos.multi_hand_landmarks:
```

```
        for hand_landmarks in resultado_manos.multi_hand_landmarks:
```

```
            mp_drawing.draw_landmarks(frame, hand_landmarks,  
mp_hands.HAND_CONNECTIONS)
```

```
            if not jugando:
```

```
                tiempo_inicio = time.time()
```

```
                jugando = True
```

```

jugada_jugador = None

resultado = ""

# Después de 5 segundos, captura la jugada

if jugando and time.time() - tiempo_inicio > contador and jugada_jugador is
None:

    jugada_jugador = detectar_jugada(hand_landmarks)

    jugada_computadora = random.choice(opciones)

    if jugada_jugador in opciones:

        resultado = determinar_ganador(jugada_jugador, jugada_computadora)

    else:

        resultado = "Gesto no válido"

if jugando:

    tiempo_transcurrido = int(time.time() - tiempo_inicio)

    tiempo_restante = max(0, contador - tiempo_transcurrido)

    cv2.putText(frame, f"Tiempo: {tiempo_restante}", (10, 40),

        cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

    if jugada_jugador:

        cv2.putText(frame, f"Tú: {jugada_jugador}", (10, 80),

            cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 2)

        cv2.putText(frame, f"PC: {jugada_computadora}", (10, 120),

            cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 255), 2)

        cv2.putText(frame, f"Resultado: {resultado}", (10, 160),

```

```
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
```

```
cv2.putText(frame, "Presiona ESPACIO para volver a jugar", (10, 200),
```

```
cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 1)
```

```
cv2.imshow("Piedra, Papel o Tijera", frame)
```

```
key = cv2.waitKey(1) & 0xFF
```

```
if key == 27: # ESC
```

```
    break
```

```
elif key == 32: # Espacio
```

```
    jugando = False
```

```
    jugada_jugador = None
```

```
    resultado = ""
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```

