Documentación

Desarrollo de Interfaces

Programa: Proyecto reconocimiento Facial

Autor: Cristina Fuster García

# **1. Documentación de usuario final:**

## **1.1. Guía de usuario**

En este proyecto utilizamos Mediapipe que es una biblioteca de Python de vision artificial ,que reconoce los movimientos de lacamara y transmision en tiempo real. Incluye reconocimientos de manos, deteccion facial, la nariz y control del curso con la punta de la nariz.

Funcionalidades Principales

reconocimientos de manos, detecta y traza los landmarks (puntos clave) con las coordenadas detallasdas de los dedos indice y otros.

Control del cursor

Pasos para la funcionalidad

1. Enseñar las manos la camara para que se dibujen.
2. Detección facial y seguimiento de la nariz

Consejos de uso: que la iluminacion sea la adecuada y evitar los movimientos rápidos para que el seguimiento sea mas preciso.

## **1.2. FAQ (Preguntas frecuentes)**

¿Que hacer si no se detecta las manos o el rostro?

Asegurarnos de que la camara este en funcionamiento y que el entorno esta iluminida.

¿Como ajustar la sensibilidad?

modificar la variable de sensibilidad.

¿Qué resolución de la cámara se recomienda?

En este proyecto está las resoluciones estandares 640x480

# **2. Documentación técnica para desarrolladores**

## **2.1. Introducción al software**

Implementar reconocimiento avanzado de puntos clave corporales y usar esos puntos para funcionalidades de control del cursor.

Características Principales

* Detección de landmarks en manos y cara
* seguimientos de puntos específicos
* Control del cursor con los movimientos de l cabeza

Tecnologías Utilizadas

* MediaPipe: para la detección y seguimiento de puntos claves
* opnCV: para las capturas de video e imágenes.

## **2.2. Arquitectura del sistema**

El usuario interactúa con la aplicación a través de la camara, obtiene los fotogramas en tiempo real, mediapipe los procesa esos datos y detecta los landmarks y los utiliza para tareas especificas como trazados y control de cursor

Y la interacción con e sistema utilizando PyAutoGUI mueve el cursor según las coordenadas.

Flujo de Trabajo:

1. Captura un fotograma de video
2. Procede e fotograma para deetectar landmarks
3. realiza acciones basadas en los landmarks detectados.

# **3. Guías de instalación y despliegue**

## **3.1. Requisitos previos**

Sistema Operativo compatible con python

Python

bibliotecas:

* Mediapipe
* ponCv-python
* Pyatogui

## **3.2. Instrucciones de instalación**

* Instrucciones de instalacion , haber instalado las dependencias correspondientes que están arriba

# **4. Mantenimiento y soporte**

## 

## 

## 

## **4.1. Manejo de errores**

Que la imagen no aparezca, significa que la ruta no está correctamente cargada.

## **4.2. Registro de cambios (changelog)**

verion 4: estuvimos trabajando la version pero sin workers de intermediario, solo convertimos la foto c¡de color en gris con el Math.roud-

versión 5: capturamos la fecha y la hora con los milisegundos con date() y getTime()

versión 6: empecemos a trabajar con los workers en este caso en concreto solo teníamos a uno.

version 7: obtener coordenadas

ersion 10: ejecutar click al juntar los dedos

**4.3. Soporte y contacto**