UNIVERSIDAD DE SALAMANCA

Anexo 4 – Manual técnico del programador

ESCUELA POLITÉCNICA SUPERIOR DE ZAMORA

APLICACIÓN DE ACCESIBILIDAD MEDIANTE SEGUIMIENTO OCULAR

Autor: **Jorge Enrique González Gonzalo**

Tutor: **Jesús Ángel Román Gallego**

Departamento de Informática y Automática

Fecha de Adjudicación: 17/11/2023

Fecha de presentación: Febrero 2024

ÍNDICE

Lista de abreviaturas

1. **TFG**

   Trabajo fin de grado

2. **XAML**

   eXtensible Application Markup Language

3. **YOLO**

   You Only Look Once

Lista de ilustraciones

1. Introducción

En este anexo se busca ayudar a los programadores, detallando la estructura de ficheros y directorios que se han usado en la aplicación, así como el funcionamiento principal que tiene cada directorio o archivo mencionado.

Para la explicación se parte tomando como directorio raíz la carpeta "Eye_tracker_WPF_app" la cual contiene dos carpetas principales, "Eye_tracker_WPF_app", que contiene la aplicación completa, y "Yolo5_150epocas_new_dataset", la cual contiene otro seguidor de ojos que se comentara resumidamente más adelante.

2. Manual del programador

Para comenzar con la explicación se inicia con la carpeta “Eye_tracker_WPF_app”. Dentro de esta carpeta se encuentran los archivos y directorios que hacer funcionar la aplicación.

2.1. Buttons

Esta carpeta contiene un archivo de C# que contiene una clase definida para los botones que se podrán reproducir. En esta clase se leen los datos guardados en el archivo “Buttons.json” y se definen funciones para añadir o eliminar botones.

2.2. Commands

En esta carpeta se define un ICommand dentro de “RelayCommand.cs”, implementación que puede exponer un método o delegado a la vista. Este actúa como una manera de enlazar comandos entre el modelo de vista y los elementos de la interfaz de usuario. [1]

Este archivo permite la conexión entre la vista “ButtonsView.xaml” y el modelo “ButtonsModel.cs”, se hablará de ambos archivos más adelante.

2.3. Models

En esta carpeta se define la clase “ButtonsModel” en ella se especifica el contenido que va a almacenar cada botón que puede ser reproducido.

- Id: Hace referencia al id característico de cada uno de los botones
- Texto: Hace referencia al texto que se reproduce del botón

2.4. Resources

Esta carpeta contiene diversos recursos los cuales se explican con más detenimiento a continuación.

2.4.1. Ajustes.py

En este archivo se define la aplicación para modificar los ajustes del seguidor de ojos, estos datos se toman y almacenan en el archivo “config.json”.

2.4.2. Botones.py

Script de Python que crea una conexión mediante un socket para comunicarse con la aplicación principal y poder enviar y recibir datos.

Estos datos son el texto por reproducir en audio y la respuesta de que se ha reproducido con éxito.

2.4.3. Buttons.json

Este archivo se usa para almacenar los datos de los botones que se pueden reproducir, estos datos son iguales a los definidos en el archivo dentro de Models.

2.4.4. calibracion.py

En este archivo se define un programa similar al seguidor de ojos con la diferencia de que se ejecuta en primer lanzamiento para medir el movimiento máximo del usuario, estos datos los almacena en "coordenadas.csv" y después toma las coordenadas máximas y mínimas registradas del usuario y las almacena en "coordenadas_max_min.csv".

2.4.5. config.json

Este archivo define diferentes parámetros de la configuración interna del seguidor de ojos.

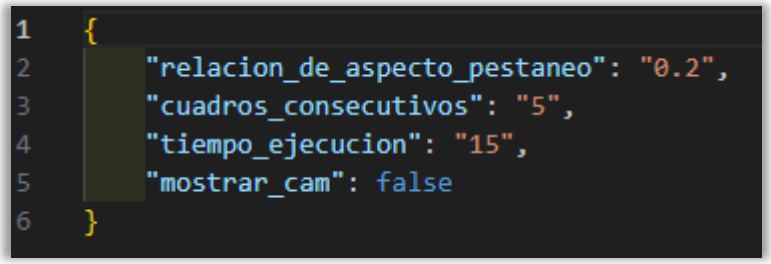

Ilustración 1: config.json

- relacion_de_aspecto_pestaneo: Hace referencia a la relación de aspecto del pestañeo, lo que significa, que relación debe haber como mínimo para que la aplicación interprete que se tienen los ojos

cerrados. Cabe recalcar que cuanto menos sea este número, más hay que cerrar los ojos para hacer clic.

- cuadros_consecutivos: hace referencia a los cuadros registrados por el programa consecutivos que deben transcurrir para realizar el clic.
- tiempo_ejecucion: Define el tiempo en segundos que se ejecutará el programa de calibración.
- mostrar_cam: Es un booleano que determina si se mostrará la cámara durante la ejecución o no.

2.4.6. coordenadas.csv

Este archivo almacena los datos comentados en calibracion.py.

2.4.7. coordenadas_max_min.csv

Este archivo almacena los datos comentados en calibracion.py.

2.4.8. EyeTracker.py

Este archivo contiene el seguidor de ojos, lee los datos tanto de configuración de config.json como los de coordenadas_max_min.csv. Después carga el predictor de puntos de referencia faciales shape_predictor_68_face_landmarks.dat, inicia el reconocimiento y lo muestra en caso de estar definido así en la configuración. Calcula el punto de referencia que se usa para el movimiento del ratón, y la relación de aspecto de los ojos para los clics.

2.4.9. shape_predictor_68_face_landmarks.dat

Predictor de puntos de referencia faciales.

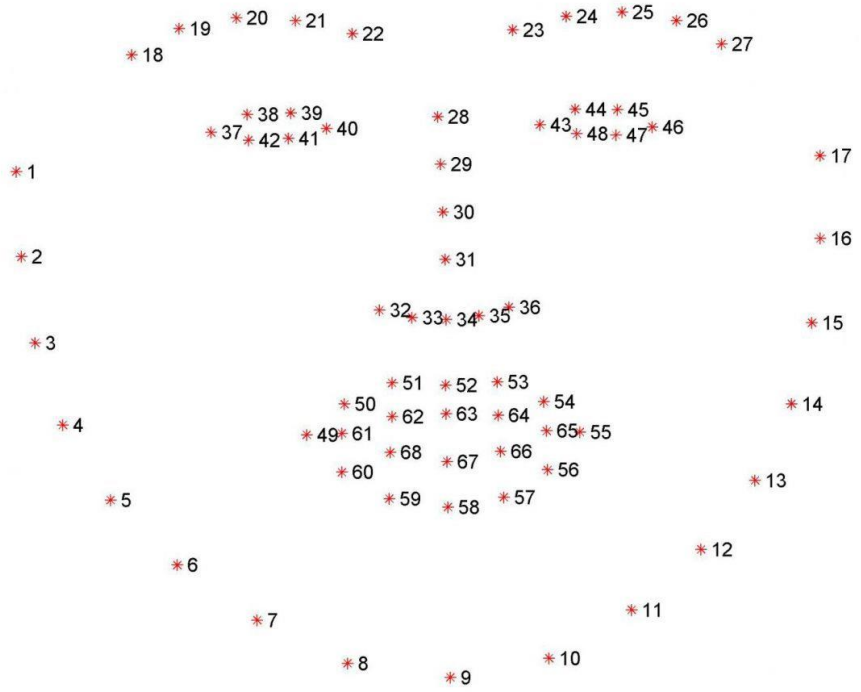

Ilustración 2: shape_predictor_68_face_landmarks

Pese a la cantidad de puntos que se muestran en este **TFG** únicamente se usan los puntos correspondientes a los contornos de los ojos.

2.4.10. styles.xaml

Este archivo define diferentes estilos que se usan en la creación de la vista de la aplicación.

2.5. ViewModels

Esta carpeta contiene dos archivos, el primero es "ViewModelBase.cs", que define características base para los viewmodels que se puedan crear. El otro archivo es "ButtonsViewModel.cs" el cual define las conexiones entre la vista y el modelo.

2.6. Views

Contiene la vista de la aplicación desarrollada en **XAML**.

2.7. App.xaml

Importa los datos de styles.xaml al programa.

2.8. Eye_tracker_WPF_app.exe.lnk

Acceso directo que ejecuta la aplicación. Se ha ubicado un acceso directo en este directorio para facilitar el acceso.

2.9. MainWindow.xaml

Define el tamaño de la ventana y carga la vista principal.

Por otra parte, está el directorio "Yolov5_150epocas_new_dataset", en esta carpeta se encuentra un modelo de seguimiento ocular entrenado con **YOLO**, pero actualmente no está aplicado al TFG debido a la bajada de rendimiento en el programa en caso de usarlo.

Referencias

[1] Microsoft, «RelayCommand y RelayCommandT<>,» [En línea]. Available: https://learn.microsoft.com/es-es/dotnet/communitytoolkit/mvvm/relaycommand. [Último acceso: 03 Febrero 2024].