Random Whispers.

1. Resumen ejecutivo.

El objetivo de este proyecto es desarrollar una aplicación de Android que se activará mediante el detector de movimiento de la cámara frontal de una tablet. Al detectar movimiento, la aplicación reproducirá un vídeo aleatorio de una lista predeterminada, garantizando que no se repita el mismo vídeo dos veces seguidas. Esta aplicación será utilizada en los pasillos de IMMUNE.

2. Objetivos del proyecto.

Proporcionar un elemento interactivo en los pasillos de IMMUNE, con la capacidad de detectar el movimiento de los transeúntes y reproducir un vídeo en respuesta a dicho movimiento. El sistema debe garantizar que el vídeo seleccionado es siempre diferente del último reproducido.

3. Requisitos del sistema.

La aplicación será desarrollada para tablets / móviles con sistema operativo Android. Requerirá acceso a la cámara frontal y a los archivos de vídeo almacenados en el dispositivo.

4. Arquitectura del sistema.

La aplicación tendrá dos componentes principales: un detector de movimiento y un reproductor de vídeo. El detector de movimiento será responsable de capturar imágenes de la cámara frontal y detectar cambios significativos. Cuando se detecte un cambio, se activará el reproductor de vídeo que seleccionará y reproducirá un vídeo aleatorio de la lista.

5. Casos de uso

Caso de uso 1: Al iniciar la aplicación, verifica si tiene los permisos necesarios para acceder a la cámara. Si no los tiene, solicita los permisos al usuario.

Caso de uso 2: Una vez concedidos los permisos, el sistema inicia la cámara y se prepara para detectar el movimiento a través de la cámara frontal. Al detectar un movimiento significativo, selecciona un vídeo aleatorio de la lista y lo reproduce.

Caso de uso 3: Al finalizar un vídeo, el sistema se detiene en el último fotograma, permaneciendo en espera hasta que detecte un nuevo movimiento.

Caso de uso 4: Si la aplicación pasa a segundo plano (por ejemplo, al cambiar a otra aplicación), el sistema desvincula la cámara. Cuando la aplicación vuelve a primer plano, la cámara se vuelve a iniciar.

Caso de uso 5: En caso de un error durante la vinculación con la cámara, el sistema registra el error y muestra un mensaje al usuario.

6. Diseño de la interfaz de usuario (UI).

La interfaz de usuario ha sido diseñada para centrarse en la reproducción de videos en respuesta al movimiento detectado. La UI consiste en dos componentes principales:

1. VideoView: Este componente se utiliza para mostrar los videos seleccionados. Cuando se detecta movimiento, se reproduce un video aleatorio de la lista predeterminada. El VideoView ocupa toda la pantalla y se ajusta automáticamente al tamaño del video para una visualización óptima.

2. PreviewView: Este componente muestra la vista previa de la cámara frontal. Permite a los usuarios ver en tiempo real su imagen capturada por la cámara y detectar el área donde se activa el detector de movimiento.

7. Explicación de las funciones.

- **onCreate()**, se llama a initActivity(), que verifica si la aplicación tiene permiso para acceder a la cámara. Si es así, se inicia la cámara; de lo contrario, se solicita el permiso.

- **startCamera()** inicia el proceso para vincular la cámara con la vista previa (PreviewView) y el análisis de imágenes (ImageAnalysis).

- **bindPreview()** vincula la vista previa y el análisis de imágenes a la cámara. Configura el ImageAnalysis para conservar sólo el último frame y establece un analizador para convertir cada frame a un objeto Mat de OpenCV, detectar el movimiento y clonar el frame para su uso en la siguiente comparación de movimiento.

- **convertImageToMat()** convierte un objeto Image de Android en un objeto Mat de OpenCV. Esto implica obtener los tres planos de color (Y, U y V) de la imagen, copiar sus datos en un array de bytes en formato NV21, poner esos datos en un objeto Mat y finalmente convertir el espacio de color de YUV a BGR.

- **detectMotion()** compara dos frames para detectar el movimiento. Aplica una serie de transformaciones a los frames (conversión a escala de grises, desenfoque gaussiano, cálculo de la diferencia absoluta y umbral) para obtener una imagen binaria que representa las áreas de cambio. Luego, si la cantidad de cambio supera un cierto umbral y ha pasado suficiente tiempo desde el último cambio detectado, elige un video de la lista de videos y lo reproduce en un VideoView. Cuando el video termina, se detiene y muestra el último frame.

- **onPause()** y **onResume()** se aseguran de que la cámara se desvincula cuando la aplicación se pone en pausa y se vuelve a vincular cuando se reanuda.

8. Pruebas.

Se realizaron pruebas unitarias en los componentes individuales del detector de movimiento y del reproductor de vídeo para asegurar su correcto funcionamiento. Después, se hicieron pruebas de integración para asegurarse de que ambos componentes interactúan correctamente. Finalmente, se realizaron pruebas de sistema completo para verificar que la aplicación cumple con los requisitos y objetivos del proyecto.

9. Planes para futuras iteraciones / mejoras.

En el futuro, podríamos considerar la implementación de reconocimiento facial para adaptar los vídeos que se muestran basándose en quién pasa por delante de la tablet. También podría considerarse la posibilidad de actualizar la lista de vídeos a través de la red, permitiendo una mayor variedad y actualizaciones más frecuentes.

10. Conclusiones.

En resumen, el desarrollo de la aplicación "Random Whispers" ha sido un ejercicio interesante en la creación de interacciones dinámicas y en tiempo real con los usuarios. El uso de bibliotecas de procesamiento de imágenes, como OpenCV, ha sido de gran utilidad para implementar un detector de movimiento preciso y eficiente. Además, cabe destacar la importancia de gestionar adecuadamente los permisos de usuario, especialmente el acceso a la cámara, para garantizar un funcionamiento fluido de la aplicación. Asimismo, la importancia de manejar adecuadamente el ciclo de vida de la aplicación, en particular al vincular y desvincular correctamente los recursos según el estado de la aplicación.