

Entrega Final - Documentación

Título: AR Filmin

Autor: Rafael Fernández Nevado. Grupo A2

Descripción general de la aplicación:

Se trata de una aplicación para ordenador de realidad aumentada que funciona como filmoteca o cartelera para mostrar información acerca de películas de una forma más interactiva.

El objetivo de esta aplicación y su utilidad es la de tener una biblioteca de películas donde guardes las películas que ya has visto o las que vayas a ver más tarde, además la aplicación guardará información sobre cada película como el reparto, o la sinopsis, y también la posibilidad de ver el trailer de la película en realidad aumentada.

Esta aplicación estaría diseñada especialmente para gente cinéfila que quiera tener su propia biblioteca de una forma más interactiva en vez de con una pantalla y escribiendo. Además, podría ser útil para gente mayor que no sepa bien cómo usar las nuevas tecnologías ya que con solo hablar podría usarla.

Papel que juegan las tecnologías involucradas:

Reconocimiento e identificación de imágenes: La app tendrá un registro de los usuarios mediante el reconocimiento facial de las personas que la usen ya que cada uno de ellos tendrá su propia filmoteca con las películas pendientes para ver o las películas que ya haya visto entre otras cosas

Reconocimiento de voz: Se le podrá dar a la aplicación una serie de palabras que reconocerá para mostrar la diferente información que tiene cada película o también para ir cambiando entre películas.

Realidad aumentada: A través de marcadores o del propio cartel de la película, se resaltará la imagen en realidad aumentada del cartel y además se mostrará en realidad aumentada la información que se solicite mediante la voz.

Implementación final

La aplicación comienza con una interfaz básica en la que se le da a un botón para iniciar el programa, en ese momento la aplicación abrirá la cámara y hará un reconocimiento facial para ver si estás registrado o no. Si no estás registrado te pedirá que introduzcas un nombre de usuario para registrarte. Una vez dentro, al ser nuevo usuario te preguntará sobre tu edad (realmente no es tu edad concreta si no si eres público infantil o adulto).

Según lo que hayas elegido, tendrás disponibles una serie de películas de las cuales podrás ver información como la portada de cartelera o la sinopsis en realidad aumentada de cada una de las películas que tenemos disponibles, además de poder ver el trailer de la película en el caso de decir “tráiler” por comando de voz. Cabe destacar que la única interacción teclado ratón que hay con la aplicación es al iniciar y al registrarse, todo lo demás es a través de comandos de voz.

Decisiones y comentarios sobre la práctica

En cuanto a la aplicación, tenemos un archivo `camara.py` básico para calibrar la cámara, pero esto no es relevante en la toma de decisiones del proyecto.

Lo primero que hay que comentar es sobre las librerías que he usado y por qué las he usado.

Obviamente la primera librería a nombrar es `cv2` para usar todo lo relacionado con OpenCV, los Arucos, y la reproducción de imágenes en realidad aumentada además de `numpy` para el cálculo de las coordenadas de los arucos para establecer la imagen correctamente dentro de estos mismos.

En cuanto al reconocimiento facial he usado la librería “`face_recognition`” ya que es la recomendada en la asignatura para esto y la he usado para la función de registro e inicio de sesión del usuario. Este apartado es bastante sencillo pero lo he hecho así adrede para que todo sea mucho más rápido y que no haya complicaciones a la hora de iniciar la aplicación.

Para el reconocimiento de voz he usado la librería “`speech_recognition`” con la que puedes utilizar diferentes APIs de reconocimiento de voz, entre ellas la de google que es la que yo he usado ya que tenía entendido que funcionaba bastante bien y me ha sorprendido bastante lo bien que funciona de verdad. Además, para que esto funcione correctamente sin fallos a la hora de la realidad aumentada, he tenido que usar la librería “`concurrent.futures`” para la creación de hebras concurrentes al programa para poder tener la ejecución de la aplicación a la vez que se usa el reconocimiento de voz.

Para la interfaz de usuario he usado `tkinter` que se usa específicamente para interfaces gráficas y lo he usado porque es muy intuitivo, fácil de usar y para lo que necesito no me hace falta nada especialmente potente y avanzado. Además para la interfaz de usuario he usado “`pynput.keyboard`” para poder introducir el nombre cuando se registra un usuario.

La reproducción de video es uno de los apartados que se me complicaron un poco más debido a las librerías disponibles que hay. Empecé usando la propia librería de OpenCV pero cuando me di cuenta, el audio de los videos no se reproducía y no sabía cómo arreglarlo así que opté por usar python-vlc que tampoco funcionaba correctamente y además era más complicada de usar. Por último encontré una librería llamada “moviepy.editor” con la cual si conseguí implementar todo correctamente y en perfecto funcionamiento.

Por último para la base de datos he usado un archivo muy sencillo de json donde se irán guardando los diferentes usuarios que se registren en el momento.

Sobre las funciones, tenemos algo también sencillo pero bastante funcional con las funciones de registro con la cual se guarda una foto del usuario y cada vez que quiera iniciar sesión (se usa la función de inicio de sesión “login”) se comparará esa foto con lo que la cámara esté viendo en ese momento y si se ve a la misma persona, dará el visto bueno al inicio de sesión.

Luego tenemos una “funcionEdad” que usamos cuando nos registramos para saber cuales son las películas que tenemos disponibles donde se guardaran las películas con sus características (cartel, sinopsis, trailer). Al principio pensé en hacerlo de manera que las películas fueran una tabla diferente dentro de la “base de datos” pero me di cuenta que tampoco era especialmente necesario y lo hice más sencillo, además de que a la hora de usar la “funcionPelículas” se podía complicar más la implementación si lo hacía con tablas diferentes

Seguidamente usamos la “funcionPelículas” para según el comando por voz veremos unas cosas u otras en el ArUco o en la pantalla. Esta función guardará en memoria todos los atributos de una película según la que se haya escogido, para así poder interactuar con ello mediante el reconocimiento de voz y los ArUcos. Esta viene acompañada de una función play_trailer usando la librería “moviepy.editor” mencionada anteriormente.

Por último tenemos la “funcionArucos” que básicamente es para la lectura de los arucos, con las imágenes que queramos mostrar y una última parte que es la interfaz la cual ya he explicado anteriormente que ha sido algo básico, sencillo y sin mucha complicación.

Comentar sobre la práctica que he tenido muchos errores a la hora de implementar librerías por problemas con “pip” o con las versiones de python, empezando por la 3.12 hasta llegar a la 3.7 la cual es la única con la que ha funcionado todo correctamente. A la hora de implementar librerías, pip instalaba los archivos que no eran, o simplemente no funcionaba lo que instalaba por lo que tenía que ir buscando librería por librería en internet y dar con la versión correcta para instalarla manualmente (esto me ha pasado con CMake, dlib y PyAudio, que es una librería adicional a moviepy).