Práctica:
Aplicación interactiva para el aprendizaje sobre elementos del Sistema Solar

Alumno: Álvaro Rodríguez Gallardo.
Asignatura: Computación Ubicua e Inteligencia Ambiental.
Subgrupo: A.2
Profesor: Antonio Bautista Bailón Morillas.

Índice

1- Información general del proyecto

La aplicación será un software interactivo mediante el cual el usuario podrá aprender cuestiones básicas sobre los elementos del Sistema Solar (aunque si desea, solo haciendo pequeñas modificaciones en el código, se podrá escalar a otros cuerpos celestes). En general, la aplicación tratará sobre cuestiones educativas, uno de los campos en los que la realidad aumentada puede tener un papel fundamental, usando el resto de tecnologías para poder hacer que la aplicación sea lo más interactiva posible.

Desde un punto de vista general, se realizará una sola aplicación, que usará un marcador. Tras escanearlo, aparecerá una especie de menú, en el cual distinguiremos: planetas, estrellas y satélites. En el caso de los tres primeros, el usuario tendrá una lista disponible con los planetas y/o estrellas disponibles para su visualización en realidad aumentada, acompañado de un cuadro de información asociada. Cuando el usuario quiera terminar, dirá ‘Atrás’ y volverá a la lista anterior.

Las plataformas en la cuales se usará la aplicación será en dispositivos móviles, en un principio solo para aquellos que tengan un sistema operativo Android. Sin embargo, para el testeo de la aplicación se usará la webcam de un ordenador con sistema operativo Windows.

Está dirigida a todos los públicos, si bien es cierto que los principales grupos de personas que estarían interesados en usar la aplicación serían grupos escolares, ya que la aplicación tiene fines educativos.

2- Interacción hombre-máquina implícita

Realidad aumentada.

Consistirá principalmente en la visualización de la lista de cuerpos celestes disponibles en función de qué elija el usuario, además de la recreación del cuerpo celeste escogido (si tiene algún satélite, se procurará representarlo girando alrededor del planeta.

En un principio se usará la propia ubicación del marcador para situar el origen de coordenadas (aunque esto puede variar en el momento en que se hagan las pruebas de la aplicación).

- En el caso de la lista, el origen de coordenadas será el centro del rectángulo en el que vendrá dada la lista de nombres.

- En el caso de un cuerpo celeste en específico, el origen de coordenadas será el centro de la esfera. Se supone que la esfera será 'perfecta', es decir, sin perturbaciones. Si así se considera, se intentará poner la órbita de un satélite (del que se tenga modelo guardado), que consistirá en seguir la **parametrización de la circunferencia, (cos(rt),sen(rt))** donde r>0 es el radio de órbita, en función del tiempo t>=0. Se iniciará a t=0, y se usará el reloj del ordenador para que parezca un movimiento continuo. Poner a t=0 va a depender de una variable booleana que indica si se entra por primera vez a visualizar el cuerpo celeste.

Procesamiento de lenguaje natural.

Principalmente se usará en las tres primeras opciones descritas en el primer punto. Existirá una lista de cuerpos celestes disponibles, y el usuario dirá cuál quiere visualizar, momento en el cual aparecerá el cuerpo pedido junto a su descripción. Además, si el usuario quiere volver a la lista, bastará con decir 'Atrás'.

Si el usuario menciona un cuerpo celeste que no existe en la lista, el sistema no hará nada. De igual manera, si en la visualización del cuerpo celeste menciona cualquier palabra distinta a 'Atrás', el sistema no hará nada.

La manera óptima de hacerlo será crear un hilo que ejecutará, en bucle infinito, una función para detectar qué dice el usuario. Si menciona una palabra, esta se almacena en una cola, variable global, a la cual accederá el proceso principal cada vez que deba comprobar si hacer algún cambio en la aplicación. Esta manera de hacer el programa será para no ralentizar en exceso la ejecución.

3- Consciencia de contexto

Tratamiento de imágenes

En primer lugar, para iniciar la aplicación el usuario deberá hacer que la cámara detecte un marcador cuya imagen está almacenada en ‘imgs’. Será importante que la cámara lo siga detectando, ya que la posición del mismo (su centro aproximado) se usará como sistema de referencia para construir los elementos de la aplicación.

Por otro lado, se dará la opción al usuario (durante la ejecución de toda la aplicación) de hacer movimientos en pos de interactuar con la aplicación. El único movimiento que interpretará la aplicación será el de hacer zoom (si el tiempo lo permite, se intentará implementar la posibilidad de poder mover, en el plano, la escena, aunque no es seguro pues aquí el centro desde el que se construye varía, tal vez usando una variable global que guarde la posición en la que se ha movido la mano, si se ha hecho, es decir, el sistema de referencia será (coordX+movimiento_X,coordY+movimiento_Y), con movimiento_X, movimiento_Y variables globales inicializadas a 0.0, que indican el desplazamiento hecho por el usuario con un dedo, y (coordX,coordY) es el centro aproximado calculado por la imagen detectada por la webcam). El zoom se interpretará como dos dedos separándose (o juntándose) y, si procede, el movimiento por el plano de la escena será con la detección de un dedo). Un posible problema de este último movimiento es que algún objeto salga de la escena. Se procurará recortar el objeto para solo mostrar parte de él, aunque también se puede optar por eliminar la visualización.

4- Resultados

De manera general, la aplicación consiste en la superposición de objetos en 2D encima de un frame captado por cámara. En primer lugar, se muestra por pantalla el frame captado hasta que consigue detectar el siguiente marcador:

Tras captar el marcador, se calcula de manera aproximada el centro del marcador captado, el cual se va a tomar como sistema de referencia para la construcción de los elementos de realidad aumentada.

Primero se muestra el tipo de cuerpo celeste que se quiere observar, y mediante el reconocimiento de voz el usuario dirá a cuál ir. Ídem en el planeta, estrella o satélite a observar. Cuando el usuario seleccione un cuerpo celeste, se mostrará la información, obtenida a partir de varias funciones auxiliares (no se ha completado el uso de una base de datos, pues no se contempla que el programa pueda ser escalable).

En lo que respecta a la construcción de los elementos de realidad aumentada, si el objeto es un ‘grid’ para mostrar la información, se toma la esquina superior izquierda (tomando como referencia el centro del marcador) y se construye una tabla, con tantas divisiones como elementos de un array se quieran colocar. En el caso del cuerpo celeste, se ha optado por crear con OpenCV un círculo blanco, con centro calculado a partir del centro del marcador y radio tomado de forma aproximada (no a escala real, ni proporciones reales, puesto que no cabía en la pantalla del dispositivo), sobre el que se ha aplicado la textura correspondiente, almacenada en una carpeta de la aplicación. Por último, si una figura sale de la pantalla, se “aplasta” para evitar el intento de acceso a partes de la pantalla que no existen y, por ende, deje de funcionar la aplicación. En el caso de los satélites orbitando, se termina de eliminar el asteroide y no continua la animación hasta que el usuario mueve un poco la cámara y permite que siga la animación.

Para el reconocimiento de voz se ha usado la biblioteca Speech Recognition, y se han lanzado dos procesos: uno ejecuta el bucle principal de la aplicación, y otro se encuentra ejecutando constantemente una función en la que se accede al micrófono del ordenador, para poder captar las palabras dichas por el usuario. La comunicación entre sendos procesos se realiza mediante el uso de una cola, variable global, en la cual se almacenan las palabras reconocidas por el proceso de escucha, y según capte el proceso principal la palabra, cambiar el comportamiento del programa (se usa una cola pues usa un estilo FIFO). Entre las palabras clave para cambiar el funcionamiento de la aplicación se encuentran:

- ‘Terminar’: La aplicación finalizará cuando el usuario diga tal palabra, dando fin a los dos bucles de los procesos y haciendo ‘join’ cada uno.
- ‘Atrás’: Si el usuario quiere volver hacia el menú anterior a la situación en la que se encuentra, será posible si menciona tal palabra.
- ‘<Nombre cuerpo celeste>’: Si está en el menú correspondiente, podrá visualizar un modelo en 3D y la información del cuerpo celeste en cuestión.

Respecto al uso de la biblioteca, se han encontrado problemas a la hora de captar ciertas palabras. A veces el usuario dice ‘terminar’, pero capta ‘terminal’, e incluso en el caso del satélite Fobos, la biblioteca ha sido incapaz de captar el mismo mediante el micrófono, aunque haciendo inferencia en las palabras que más capta cuando se menciona este nombre, se ha llegado a la conclusión que siempre van a acabar en ‘os’.

Pasando a las animaciones, se ha usado MathPlotLib para la elaboración de algunas animaciones simples, las cuales consiste en el movimiento de traslación de los satélites en torno al planeta, si procede. Por ejemplo, la Luna en torno a la Tierra o Deimos y Fobos en torno a Marte (si bien en este último caso, se ha optado por suponer que sendos asteroides son simétricos respecto al centro de Marte, con idénticas velocidades). Se ha usado una variable global que indica el ángulo que va a tomar respecto al centro del planeta y un radio mayor al del propio planeta, expresado en radianes, y en cada iteración aumenta en uno (lo cual no supone problema, pues las funciones seno y coseno son dos pi periódicas).

Por último, mencionar el uso de la biblioteca Mediapipe Hands, para poder ofrecer al usuario la posibilidad de hacer o deshacer zoom (siempre dentro de unos límites ya establecidos) con el movimiento de los dedos pulgar e índice de una mano. En dos variables globales almacenamos los datos de las posiciones antiguas de estos, y con las posiciones captadas en el frame en curso, se comprueba si se han separado (hacer zoom) o se han acercado (deshacer zoom). Debido a varios problemas con la distinción de dedos, se ha suprimido la posibilidad de mover los objetos con un dedo.

5- Comentarios sobre la aplicación

Respecto a la idea que se pensó en un primer momento, no ha variado mucho el producto final excepto en la idea de haber implementado una animación del sistema solar que sea fidedigna a lo que ocurre en la realidad. En un primer momento se pensó que solo accediendo al reloj del ordenador y usando ciertas parametrizaciones de la elipse y la circunferencia bastaría para realizar la animación, pero se desistió puesto que para dibujar elementos sobre el frame, OpenCV no acepta floats, y si se hubiese optado por redondear dicho float, si se llegase a un momento en que redondease a la baja, no se movería el cuerpo, además de que la posición del centro no hubiese sido creciente, sino que hubiese ido variando (pues queremos que las trayectorias sigan objetos del plano, no funciones de una variable estrictamente crecientes).

En lo que respecta al reconocimiento de voz, ya se comentó anteriormente los problemas que daba en ciertos momentos la biblioteca: hay veces que capta la palabra que dice el usuario, pero con una letra cambiada, y por consecuencia el programa no hace nada.

Para terminar, se desiste de la opción de poder mover los elementos con el dedo pues con el simple hecho que la cámara detectase los dedos para hacer zoom, el factor de escala variaría, haciendo zoom o deshaciendo cuando no se quiere realizar tal acción. Se ha pensado en usar el dedo corazón para ello, pero si se evita que se capten el pulgar y el índice, incomodaría bastante al usuario.

La aplicación, pese a haber sido desarrollada en un ordenador, está pensada para móviles, por ello que deba reconocer siempre una imagen que se sostiene delante de cámara: en un móvil este inconveniente desaparece.

6- Experiencia de usuarios ajenos al desarrollo

Tras hacer que varios usuarios externos al desarrollo de la aplicación usasen la misma, se ha podido comprobar que tenían problemas, sobre todo, con el procesamiento del habla, puesto que había momentos en los que decían una palabra y el sistema no hacía caso. Tras un análisis del funcionamiento de la aplicación, se ha podido observar que la biblioteca captaba las palabras raramente, es decir, muchas palabras que decía haber captado eran otras distintas a las dichas, o con pronunciación similar. Se desconoce si este problema es del hardware, o de la propia biblioteca, por lo que para posteriores desarrollos sería interesante revisar más a fondo esta cuestión (hacer un motor de inferencia para inferir qué ha podido decir, y el significado de lo que se ha podido decir).

Los usuarios sentían, por lo general, cierta rabia al ver que por más que decían palabras, el sistema a veces los ignoraba (pues la biblioteca captaba otras palabras). Exceptuando este problema, el resto de los elementos del software no han supuesto mucho problema (si bien ha costado que ‘Mediapipe Hands’ reconozca los dedos del usuario, pues pensaba que se trataba de algo parecido a la pantalla táctil de un móvil, sin dejar que la webcam enfocase bien la mano).