**CAPITULO 2-MARCO TEORICO.**

**DEFINICIONES Y ABREVIATURAS**

**Lenguaje de Señas:**

Desde hace mucho tiempo los sistemas de signos eran utilizados no sólo por las personas sordas. Entre algunas tribus indias del norte de América eran utilizados para facilitar la comunicación debido a la cantidad de lenguas diferentes que utilizaban. Entre los primeros estudiosos de los sistemas de signos como medio de comunicación para personas sordas estuvieron Juan de Pablo Bonet ,(1573-1633) pedagogo español que escribió “Reducción de las letras y Arte para enseñar á hablar los Mudos “, libro en el que mostraba un método de comunicación mediante alfabeto de signos para las personas sordas.

A partir del alfabeto publicado por Juan de Pablo Bonet, Charles-Michel de l'Épée (1712-1789) publica un alfabeto en el que se basan los alfabetos de signos que se usan en la actualidad. Dicho alfabeto fue publicado por su sucesor el abate Sicard en la forma de “Diccionario general de Signos”.

En 1817 Thomas Gallaudet abrió una escuela de sordos en Connecticut (Estados Unidos) donde sentó las bases para la ASL, tomando como referencia la lengua de signos francesa y algunos signos utilizados por las tribus indias de Norteamérica. En 1965 William Stokoe, en el libro “A Dictionary of American Sign Language on Linguistic Principles” describió una gramática precisa para el lenguaje de signos americano.

La lengua de signos para el idioma Ingles Americano (American Sign Language,ASL) es unn lenguaje no oral, no auditivo, sino gestual y visual constituiría un nuevo campo de investigación para la ciencia lingüística, centrada, hasta entonces, en los lenguajes hablados . Utilizada en Estados unidos, el norte de México y la zona anglófona de Canadá.

**Kit de desarrollo de software (SDK**): Conjunto de herramientas de desarrollo software que permite al desarrollador crear aplicaciones para un sistema concreto.

**(6) Vuforia**

Vuforia es un SDK que permite construir aplicaciones basadas en la realidad aumentada que según su propia descripción, una aplicación desarrollada con Vuforia utiliza la pantalla del dispositivo como un "lente mágico" en donde se entrelazan elementos del mundo real con elementos virtuales como letras, imágenes, objetos 3d, etc.

Su funcionamiento se basa en la detección de ciertas imágenes usando la cámara del dispositivo (**Trackables**). Las funciones de la librería proporcionan la posición y orientación de dichos trakeables a través de una matriz de 4×4 llamada **matriz de pose.**

**Esta** plataforma de desarrollo de software pone a disposición de los programadores de aplicaciones móviles un motor de reconocimiento de imágenes muy potente, así como un amplio abanico de herramientas diseñado para permitirles dar rienda suelta a su creatividad sin que se vean obligados a preocuparse por las limitaciones de índole técnica.



El desarrollador debe disponer previamente de las imágenes que quiere reconocer con su aplicación. Las cuales deben ser subidas al Web Management System de Vuforia y, o bien se crea una base de datos de imágenes alojada en la nube que se consultará en tiempo de ejecución, o bien se descarga dicha base de datos y se incluye en la aplicación. Durante la ejecución de la aplicación, esta interacciona con el motor de Vuforia para reconocer las imágenes almacenadas en la base de datos. Cuando el motor de Vuforia detecta una imagen en la escena, se puede proyectar el contenido virtual creado por el desarrollador.

**Arquitectura**

Una aplicación de Realidad Aumentada basada en Vuforia, consta de una serie de componentes muchos de estos son singletons. Un singleton consiste en una clase diseñada para tener sólo una instancia (o un número limitado de ellas), por comodidad a la hora de emplear el lenguaje, se llamará instancia a un singleton.

• Cámara (camera, singleton): La instancia de la cámara se encarga de que cada fotograma capturado por la cámara digital se pase de forma eficiente al tracker.

• Convertidor de imágenes (Image Converter, singleton): La instancia del conversor de formato de pixel realiza la conversión entre el formato con el que trabaja la cámara a un formato adecuado para el renderizado en OpenGL ES y para el seguimiento (por ejemplo, luminancia).

• Tracker (singleton): El tracker contiene los algoritmos de visión computacional para detectar y seguir (detect & track) los objetos en los fotogramas capturados por la cámara. Basado en la imagen tomada por cámara, diferentes algoritmos se ocupan de detectar nuevas imágenes de referencia (Targets) o marcadores (Markers), y evaluar los botones virtuales (Virtual Buttons). Los resultados se almacenan en un objeto de estado que es utilizado por el procesador de vídeo de fondo y al que puede accederse desde el código de la aplicación. El tracker puede cargar múltiples conjuntos de datos (datasets), pero sólo uno puede estar activo a la vez.

• Procesador de vídeo de fondo (Video Background Rendered, singleton): La instancia del procesador de vídeo de fondo procesa la imagen capturada por la cámara que se encuentra almacenada en el objeto de estado.

• Código de la aplicación (Application Code): El desarrollador ha de inicializar todos los componentes anteriores y llevar a cabo tres pasos fundamentales en el código de la aplicación. Por cada fotograma procesado, el objeto de estado se actualiza y se llama al método de procesamiento de la aplicación. El desarrollador debe:

* Consultar el objeto de estado para los Targets y/o Markers nuevos que puedan aparecer en escena o si se actualiza su estado.
* Actualizar la lógica de la aplicación con nuevos datos de entrada.
* Renderizar la capa de Realidad Aumentada.

• Recursos de imágenes de referencia (Target Resources): Los Target Resources se crean mediante el Target Management System, que se encuentra disponible online. El dataset descargado contiene un fichero de configuración XML que permite al desarrollador configurar ciertas características de los trackables y un fichero binario que contiene la base de datos de los trackables. Estos elementos (XML y binario) son compilados por la aplicación a desarrollar en el paquete de instalación de la aplicación y los usa el SDK Vuforia en tiempo de ejecución.



Diagrama de flujo de datos en una aplicación de ejemplo

La figura muestra el diagrama de flujo de datos en una aplicación de ejemplo que usa el SDK de Vuforia. La cámara del dispositivo obtiene un frame que se convierte al formato requerido y se pasa al tracker, que puede tener distintas configuraciones. Este proyecto se ha basado en el tracker ImageTargets, es decir, reconoce un solo target o marcador en el frame. Esto es así porque la superficie plana se considera como un marcador-imagen, y suponemos que en la escena solo vamos a tener una única superficie plana dominante, por tanto no tendrá sentido poder reconocer dos marcadores, es decir, dos superficies planas distintas sobre las que proyectar nuestro modelo. El tracker obtiene la lista de marcadores que tiene que buscar de la base de datos alojada en la nube, o integrada en la aplicación, o se han podido crear los marcadores en tiempo de ejecución a partir de frames obtenidos por la cámara del dispositivo móvil. Esta última opción es la que se ha desarrollado en el proyecto. Finalmente, el tracker indica dónde debe renderizarse el modelo, y el nuevo frame se proyecta en la pantalla.

Arquitectura de la aplicación

Nuestra arquitectura no varía mucho de la arquitectura original de vofuria. la diferencia principal es la inclusión de una base de datos con la que actúa la aplicación, la cual contiene toda la información que muestra el menú y gestiona las transacciones de los pedidos realizados.



**Figura de la arquitectura de la aplicacion**

Unity Extension - Vuforia v2.8: es una librería que se instala en el programa Unity para poder trabajar con realidad aumentada.

https://developer.vuforia.com/resources/sdk/unity

http://www.formaciononlinegratis.net/como-crear-image-target-en-vuforia

Visión por computador en dispositivos móviles para realidad aumentada- Alfonso Escriche Martínez - Septiembre de 2013- Universidad de Zaragoza

Consultada el:23-ene-2014

**(4)** Modelado 3D

Strata Foto 3D CX 2

Se trata de una forma sencilla de crear y colocar texturas UV en cualquier modelo 3D haciendo uso de fotografías. Podemos usar la cantidad de fotos para crear la textura de un modelo 3D, con ajuste de calibrado automático; lo único que se necesita es una cámara digital convencional.

La forma del objeto y sus colores se extraen automáticamente de cada foto, obteniendo así información que servirá a la construcción del modelo 3D, modelo que podrás editar luego en Strata 3D CX, importar a una animación Flash, etc.

http://www.macworld.com/article/1146037/stratafoto.html Consultada el:23-ene-2014