DEFINICIONES:

API: (Application Programming Interface). Conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

FUNDAMENTOS TEÓRICOS:

La detección de gestos es un ámbito de estudio recurrente en la ciencia de la computación, haciendo uso de algoritmos matemáticos para ser capaces de detectar e interpretar dichos gestos. Enfocado, mayoritariamente, en el reconocimiento de expresiones faciales y gestos con las manos. Interpretando el lenguaje de signos [4], mediante cámaras y sensores, las máquinas consiguen ser capaces de entender e interpretar los gestos realizados por las personas, logrando de ese modo un mayor *HCI.*

Mediante la detección de gestos, algunos desarrolladores pretenden sustituir (o convivir) con el uso del teclado y del ratón.

Leap Motion es un pequeño sensor de control gestual que nos permite capturar con mucha precisión nuestras manos, incluyendo dedos, articulaciones y objetos. Esta interfaz 3D nos da la opción de interactuar con nuestro ordenador sin ni siquiera tocarlo, dejando de lado dispositivos convencionales como son el teclado y el ratón, o más modernos como las pantallas táctiles.

Algunas de las más destacadas son:

* Navegar por Internet, leer artículos, ver fotos, vídeos o reproducir música con sólo mover un dedo.
* Dibujar, pintar, diseñar con la punta del dedo. Además, se puede utilizar un lápiz real o un pincel.
* Esculpir, moldear, estirar, doblar y construir objetos 3D. Desarmar y volver a unir objetos.
* Introducir la mano en diferentes mundos, agarrar objetos, girarlos.
* Jugar a infinidad de juegos usando tan solo las manos.

El diseño de este dispositivo, consiste en un aparato con unas dimensiones muy reducidas, que se conecta mediante USB a nuestro ordenador. Después tan solo hay que configurar el dispositivo con el software específico y listo.

La API de Leap Motion viene con pocos gestos incorporados, por lo que para aplicaciones complejas es necesario que el usuario de-

fina su propio conjunto de interacciones a usar. Estamos acostumbra-dos a estados binarios de seleccionado o no, arriba o abajo. Con Leap Motion, la experiencia se define menos por los estados individuales y más por las transacciones entre dichos estados. Debe responder constantemente a los movimientos del usuario (feedback), para dar a entender que nos está escuchando y entendiendo. Esto se aleja los controla-dores tradicionales, en los que la interfaz solo cambia cuando el usuario actúa directamente sobre el dispositivo.

TECNOLOGÍAS PREDECESORAS:

El primer dispositivo de control de movimiento fue proporcionado por Nintendo Wii. Se introdujo la detección de movimiento aplicada a di-versos juegos.

Más tarde el Kinect de Microsoft amplió esta idea incluyendo el movimiento de todo el cuerpo además de eliminar cualquier elemento que se interpusiera entre la pantalla y el usuario como mandos u otros. Esto introdujo una nueva generación de sensores con control gestual.

HARDWARE:

El LEAP Motion contiene en una caja de metal de 75 mm de largo, 25 mm de ancho y 11 mm de alto los componentes hardware necesarios para su correcto funcionamiento. Entre ellos, encontramos dos cámaras, tres luces leds, un microcontrolador y un controlador USB. Este dispositivo es compatible con los sistemas operativos Linux, Macintosh y Windows.

SOFTWARE:

El procesado de los gestos y movimientos capturado por el LEAP Motion se realiza en el dispositivo al que está conectado, en un software llamado “Leap Motion Service”. Los sensores captan las distintas secuencias de movimientos en imágenes, que son enviadas al ordenador de destino. Una vez el ordenador las recibe, analiza y compensa las imágenes con la iluminación y herramientas necesarias, eliminando el “ruido” de fondo como la cabeza, muebles, luces… u otros elementos que puedan interferir en la posterior detección de gestos.

A continuación, se procede a extraer de la imagen capturada una reconstrucción en 3D de lo que realmente el dispositivo “ve”, intentando obtener los dedos y las manos de las imágenes ya “limpias” (sin el ruido de objetos de fondo). En este proceso se aplican técnicas de filtrado para asegurar una lectura del movimiento suave, sin cortes y coherente a tiempo real.

Una vez el LEAP Motion Service tiene el movimiento “limpio” guardado, lo transmite mediante un protocolo de transporte (TCP o WebSocket, dependiendo de dónde se use el dispositivo) sobre una conexión local. La librería del cliente organiza los datos obtenidos (el análisis del movimiento) en la estructura API creada por los desarrolladores, y los enlaza; de este modo se asocia el movimiento capturado con la clase o función que le corresponde en el API. Una vez vinculado el gesto grabado, y encontrada la clase o función, se procede a ejecutar la llamada a la acción vinculada.

FUNCIONAMIENTO INTERNO:

El LEAP Motion, como bien hemos explicado anteriormente, únicamente se encarga de capturar los distintos movimientos con sus sensores, no los procesa; de ello se encarga la computadora a la que se conecta. El procesado de gestos pasa por 3 etapas para llegar a ser interpretado por el ordenador: una vez las imágenes son capturadas por los sensores, se aplica una corrección de la distorsión que producen los sensores. Más tarde se aplica un modelo matemático para determinar la configuración de cada mano, junto a un algoritmo de visión estereoscópica entre cada pareja de imágenes con la finalidad de conocer la posición en el plano que representa LEAP tridimensional.

Para identificar los distintos gestos de manera correcta, cuando se detecta movimiento sobre el dispositivo, éste ilumina la zona a la que enfocan las cámaras con una luz infrarroja emitida por sus luces LEDs. Cuando esta luz refleja en las manos, los sensores incorporados en el controlador reciben esta reflexión y la almacenan en una matriz como una imagen digitalizada. Los sensores captan el valor de intensidad luminosa por cada píxel de la imagen capturada, y se guardan en un buffer. Este valor de la intensidad luminosa se cuantifica a 8 bits (256 va-lores distintos de luminosidad) para producir una imagen RAW en es-cala de grises. Cada imagen tiene una medida de 640x120 píxeles, por lo que en total hay unos 76.800 píxeles por cada imagen. Es en este momento del proceso donde el microcontrolador ajusta a la resolución adecuada la matriz con la imagen digitalizada.

Una vez ajustada correctamente la resolución, los datos de los sensores se envían directamente al driver instalado en la computadora conectada por USB. Una vez el driver tiene esta información, la analiza y trata de identificar las manos y los dedos a partir de una secuencia matemática de caracterización anatómica. También se obtiene la profundidad a la que se encuentran mediante otro algoritmo distinto, que se explicará también a continuación.

Distintos tipos de distorsiones.

Las lentes del dispositivo producen una distorsión en la imagen óptica de lo que realmente capta; se trata de una mezcla entre la distorsión de barril y la distorsión de cojín: una distorsión compleja. Esta distorsión deforma el objeto observado, y para optimizarla, LEAP Motion incorpora una opción de calibrado mediante la cual se obtiene un mapa de mallado de puntos que se superpone a la imagen capturada por cada sensor.

El dispositivo trata estos datos de la siguiente manera: cada buffer de datos de imagen que se envía al driver contiene otro buffer que contiene los datos de distorsión. Mediante una rejilla de 64x64 puntos con dos valores de 32 bits cada uno se representa la distorsión de la lente. Así, cada punto representa un rayo proyectado en la cámara, y el valor de un punto del mallado define la luminosidad de un píxel en la imagen. Se puede obtener el valor del brillo para cualquier rayo proyectado, al igual que los datos de luminosidad de todos los píxeles, pero este último dato se obtiene mediante interpolación.

La siguiente imagen nos permite explicar cómo se aplica la corrección de la distorsión compleja que crea la lente; como se puede apreciar, no es una imagen rectangular, pues al estar obtener la imagen deformada y posteriormente aplicar la corrección, se deforma su figura. La imagen se reconstruye mediante el cálculo de las pistas horizontales y verticales representados por cada píxel y se puede encontrar el valor de brillo verdadero. El valor del brillo que realmente proporciona el reflejo de la mano se puede encontrar en el mapa de calibración.

Una vez contamos con la proyección de la imagen con la distorsión corregida, el driver tratará de identificar las manos y dedos de esa secuencia. Una vez las detecte, lo primero que determinará será su posición en el sistema de coordenadas en el que trabaja LEAP. Se trata de un sistema de coordenadas cartesianas a través de técnicas de visión estereoscópicas; como su sistema de detección de gestos está basado en la visión binocular, podremos obtener distancias “fácilmente”.

El sistema de visión binocular capta dos imágenes con cierta disparidad (diferencias mínimas entre las dos fotos). Funciona de la misma manera que si las cámaras fueran nuestros ojos: al encontrarse un poco separados, la imagen que ambos captan no es la misma, pero sí muy similar. Mediante un sistema de dos ecuaciones, considerando que la distancia focal en ambas cámaras es la misma y conociendo la distancia a la que se encuentran separadas, podemos obtener la distancia a la que se encuentran las manos respecto al LEAP Motion.

API:

─ Hand:

Esta clase implementada en el API del LEAP Motion aporta información sobre las manos que detecta. Representada en base ortonormal, aporta datos como los grados de inclinación, radio de curvatura de la mano, posición relativa, probabilidad del movimiento de la mano del siguiente frame…

─ HandList:

Se trata de una lista de manos que contiene las manos detectadas simultáneamente (en Objetos hand). También existe una list para los objetos tipo finger, que es FingerList, con métodos similares en ambas clases.

─ Finger:

Aporta información (también en base ortonormal) sobre cada dedo detectado en cada una de las manos (mano a la que le corresponde, posición, inclinación, vector de dirección, anchura y longitud...).

─ Bone:

Refleja el esqueleto de la mano representada, concretamente las falanges de los dedos. La posición de la articulación, anchura, longitud o posición son algunos de los datos que se pueden extraer de esta clase.

─ Frame:

Es el conjunto de manos o dedos representados en grupo, en un único frame.

─ Listener:

La clase Listener se encarga de responder a los eventos proporcionados por el objeto Controller. Estas respuestas pueden ser configuradas y personalizadas para cambiarse por otras según el gusto del usuario. Para controlar los eventos de movimiento se instancia una subclase de Listener y se le asigna una instancia del controlador. Este controlador llama a la función establecida del Listener cuando el evento se activa.

REALIDAD VIRTUAL:

La **definición de realidad virtual** viene, naturalmente, de las definiciones para ambos "virtual" y la "realidad". La definición de "virtual" está cerca y la realidad es lo que experimentamos como seres humanos. Así 'realidad virtual' el término significa básicamente un "cuasi-realidad". Esto podría, por supuesto, significa nada, pero por lo general se refiere a un tipo específico de emulación de la realidad.

la realidad virtual es el término utilizado para describir**un entorno en tres dimensiones, generada por ordenador** que se puede explorar e interactuó con una persona. Esa persona se convierte en parte de este mundo virtual o está inmerso dentro de este entorno y mientras allí, es capaz de manipular objetos o llevar a cabo una serie de acciones.

La realidad virtual puede conducir a nuevos y emocionantes descubrimientos en estas áreas que tienen un impacto sobre nuestra vida cotidiana.

Los conceptos detrás de [la realidad virtual](http://www.vrs.org.uk/) se basan en las teorías acerca de un deseo humano desde hace mucho tiempo para **escapar de los límites del "mundo real"** al abrazar el ciberespacio. Una vez allí podemos interactuar con este entorno virtual de una manera más natural que generará **nuevas formas de interacción hombre-máquina** (HMI).

El objetivo es**ir más allá de las formas estándar de interacción**, tales como el teclado y el ratón que la mayoría de la gente trabaja con sobre una base diaria. Esto es visto como una forma antinatural de trabajo que obliga a las personas a adaptarse a las exigencias de la tecnología en lugar de al revés.

Pero un entorno virtual hace lo contrario. Se**permite a alguien para sumergirse** en un mundo altamente visual que exploran a través de sus sentidos. Esta forma natural de la interacción dentro de este mundo a menudo da lugar a nuevas formas de comunicación y la comprensión.

Educación y formación

A pasos agigantados avanza en el ámbito de la educación, aunque aún queda mucho por hacer. Las posibilidades de la realidad virtual y la educación son infinitas y traen muchas ventajas a los alumnos de todas las edades. Pocos están creando contenido para la educación, ya que toda la atención y avances se están realizando en la industria del entretenimiento, aunque muchos dan por hecho que es lo que viene en el futuro y será una pieza clave en la educación.

En estudios universitarios esta ya es usada con fines de práctica y para generar experiencia como para diseñar modelos de arquitecturas (ingenierías) o ver algunos sistemas del cuerpo humano (medicina).

Formación o entrenamiento

El uso de la realidad virtual permite entrenar a los profesionales militares en un entorno virtual donde pueden mejorar sus habilidades sin la consecuencia de entrenar en un campo de batalla.

La realidad virtual juega un papel importante en el entrenamiento de combate para los militares. Permite a los reclutas entrenar bajo un ambiente controlado donde responden a diferentes tipos de situaciones de combate. Una realidad virtual totalmente envolvente que utiliza una pantalla montada en la cabeza (HMD), trajes de datos, guante de datos, y el arma de realidad virtual que se utilizan para entrenar en combate. Esta configuración permite que el tiempo de reposición del entrenamiento sea más corto y permite una mayor repetición en un corto período de tiempo. El entorno de entrenamiento es totalmente inmersiva, permite a los soldados entrenar a través de una amplia variedad de terrenos, situaciones y escenarios.

La realidad virtual también se utiliza en la simulación de vuelo para la Fuerza Aérea donde las personas se entrenan para ser pilotos. El simulador se instalaba en la parte superior de un sistema de elevación hidráulico que reacciona a las órdenes y eventos del usuario. Cuando el piloto dirige el avión, el módulo se gira e inclina en para proporcionar retroalimentación háptica. El simulador de vuelo puede variar desde un módulo completamente cerrado a una serie de monitores de ordenador que proporcionan el punto de vista del piloto. Las razones más importantes sobre el uso de simuladores educacionales con un avión real son la reducción de los tiempos de transferencia entre la formación de la tierra y de vuelo real, la seguridad, la economía y la ausencia de contaminación. De la misma manera, las simulaciones de conducción virtuales se utilizan para entrenar a conductores de tanques en los conceptos básicos antes de que se les permita operar el vehículo real. Por último, lo mismo pasa con simuladores de conducción de camiones, en los que los bomberos belgas son entrenados para conducir de una manera que impide el mayor daño posible. A medida que estos conductores poseen menos experiencia que otros conductores de camiones, la formación de realidad virtual les permite compensar esto. En un futuro próximo, se espera que todos los proyectos similares tengan esta capacitación, incluyendo la policía.

PRODUCTOS:

Conocidos también como HMD, se distinguen fundamentalmente dos tipos: los que llevan pantalla incorporada y los que son esencialmente una carcasa destinada a que el usuario introduzca un smartphone.

En cuanto al display, solía utilizarse tecnología LCD, aunque empiezan a aparecer algunos como el Razer OSVR HDK 2, el propio PlayStation VR, o el nuevo Oculus con pantallas OLED. Mientras que algunos HMDs utilizan dos displays LCD (uno para cada ojo), otros optan por un único display con una división en el centro. Algunos tienen unas lentes colocadas entre los ojos y el display, y pueden ajustarse a la distancia de los ojos. Las lentes modifican la imagen para cada ojo, cambiando el ángulo de la imagen 2D de cada display para crear un efecto 3D, simulando las diferencias con las que se ven las cosas con un ojo respecto al otro

Otro aspecto importante de los HMDs es el campo de visión. Los seres humanos tenemos un campo de visión horizontal de unos 180º a 220º, en ocasiones más, aunque varía de persona a persona. Esta visión es monocular, es decir sólo es percibida por uno de los dos ojos. El campo de visión percibido por ambos ojos (y que por tanto vemos en 3D) es de unos 114º. Por este motivo, un campo de visión de 360º seria innecesario. La mayoría de HMDs funcionan con un campo de visión de entre 110º y 120º.

Por último, hay que destacar dos puntos: los fotogramas por segundo (FPS) y la latencia. Es imprescindible un mínimo de 60 FPS para que el ojo perciba las imágenes de manera natural y no provoque mareo. Todos los HMDs importantes superan este mínimo. El otro punto es la latencia, que ha de ser inferior a 20 ms para que el usuario no experimente una sensación de retraso entre lo que hace y lo que ve.

Oculus Rift:

Software

Los juegos y sus plataformas deben ser diseñados específicamente para funcionar correctamente con Oculus Rift. Para ello, Oculus ha creado un kit de desarrollo de software (SDK) para ayudar a los desarrolladores con la integración de Oculus Rift en sus juegos. Este incluye código, ejemplos y documentación. La integración de Oculus Rift en los juegos empezará con los PC y Smartphones, seguido más adelante por las consolas. Desde su inicio, muchos desarrolladores han estado trabajando en su integración.Algunos títulos pueden ser jugables gracias al código abierto.

PLATAFORMAS:

Al diseñar un videojuego, un desarrollador se encontrará frecuentemente teniendo que realizan comportamientos muy similares. Creando un juego desde cero cada vez que por lo tanto no ser muy productivo. Para solucionar ese problema, hay Plataformas de desarrollo que ofrecen una biblioteca de comportamientos comunes, lo que facilita a un desarrollador reutilizar las funcionalidades repetidas. Estas plataformas también ofrecen a menudo la exportación a múltiples plataformas, ahorrando tiempo al desarrollador programando el mismo juego para las diferentes plataformas.

Unity 3D:

Unity 3D es un motor de juego utilizado para desarrollar juegos complejos. Es multiplataforma, soportando la exportación a diversas plataformas como Windows, Android, iOS, Linux, Flash, PlayStation, etc. La última versión de Unity es 5.5.0 que es la que estoy utilizando en mi proyecto. Hay varias versiones disponibles según la necesidad del desarrollador o empresa. En mi caso estoy utilizando la versión estudiante ya que no necesito ninguna de las funcionalidades que de las versiones de pago.

Unity fue creado principalmente para crear juegos 3D, pero a partir de la versión 4.3 tiene soporte nativo 2D. Soporta la importación de modelos para una variedad de formatos de archivo como 3ds Max, Cinema 4D, Maya, etc. El motor de renderizado de gráfico utiliza Direct3D y OpenGL dependiendo de la plataforma de destino. La unidad está escrita en C++, y para fines de scripts soporta C#. También ofrece un IDE intuitivo y una herramienta MonoDevelop para la depuración de guiones, además un editor intuitivo que permite un inicio rápido incluso para los programadores que comienzan con el desarrollo del juego. El framework Unity3D también incluye un potente motor de física 3D nVidia PhysX Physics15, para que los desarrolladores puedan concentrarse en el juego en lugar de perder tiempo en la implementación de la física.