Proyecto Fin de Carrera

INVESTIGACIÓN DE LA APLICACIÓN DE LA TECNOLOGÍA KINECT EN ENTORNOS NUCLEARES

(Investigation on applying the Kinect Technology in Nuclear Environments)

Para acceder al Titulo de

INGENIERO EN INFORMÁTICA

Autor: Darío Muga González

Octubre - 2012

INGENIERÍA EN INFORMÁTICA

CALIFICACIÓN DEL PROYECTO FIN DE CARRERA

Realizado por: **Darío Muga González**
Director del PFC: **Miguel Sierra Sánchez**
Título: "Aplicación nuclear de la Tecnología Kinect"
Title: "Nuclear applications of Kinect Technology"

Presentado a examen el día:

para acceder al Título de

INGENIERO EN INFORMÁTICA

Composición del Tribunal:
Presidente (Apellidos, Nombre): González Harbour, Michael
Secretario (Apellidos, Nombre): Martínez Fernández, María del Carmen
Vocal (Apellidos, Nombre): Menéndez de Llano Rozas, Rafael
Vocal (Apellidos, Nombre): Sánchez Barreiro, Pablo
Vocal (Apellidos, Nombre): Sanz Gil, Roberto

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente Fdo.: El Secretario

Fdo.: Vocal Fdo.: Vocal

Fdo.: Vocal Fdo.: El Director del PFC

Agradecimientos

En el desarrollo de este proyecto de fin de carrera no me he encontrado solo, a mi lado han estado personas que me han dado ánimos o me han ayudado a salir de un bache indicándome dónde buscar cuando no encontraba la manera de solucionar un problema.

En primer ligar quiero dar las gracias a Michael González Harbour que en calidad de tutor me ha guiado a través de los trámites administrativos que he tenido que realizar para poder presentar este proyecto.

Me gustaría agradecer de manera muy especial a Migue Sierra Sánchez, director de este proyecto de fin de carrera que me ha apoyado en todo momento y cada vez que me veía me aportaba ideas o la información que encontraba sobre la tecnología que he investigado. Siempre con una sonrisa y con el mejor de los tratos me ha ayudado a seguir motivado en el proyecto durante todo su desarrollo y a hacerlo de manera agradable.

También quisiera agradecer a los compañeros de trabajo que hacían que en la empresa reinase un ambiente de compañerismo y amistad apoyándome en todo momento, riéndose de este empleado al que "han dado una Kinect para que juegue en el trabajo" mientras se levantaba de su asiento y hacía gestos raros en el aire intentando controlar su proyecto.

En un lugar especial está también mi grupo de amigos que en todo momento se han interesado por mi avance y se preguntaban cuándo me verían actuar como en las películas futuristas manejando un ordenador sin tocarlo.

Por último quiero agradecer a mi familia el apoyo que me han dado, a mis padres Fernando y Celia, a mi hermana Andrea, a mis tíos, primos y abuelos que han seguido la pista de mi proyecto durante todo el proceso y me han preguntado por ello siempre que han tenido la ocasión. Gracias.

Este proyecto supone un punto y seguido en mi vida, acabando la etapa de mi carrera universitaria y dando paso a una etapa en la que me seguiré formando y a la vez intentaré continuar mi andadura en el mundo laboral.

A todos, con el corazón en la mano, gracias.

Darío Muga.

Resumen

El objetivo del presente Proyecto de Fin de Carrera es determinar si la tecnología Kinect puede aplicarse como interfaz para una aplicación desarrollada para el entorno nuclear.

Se investigarán las posibilidades de implantación de la tecnología de tal modo que los operarios de la central puedan utilizar un sistema de monitorización sin entrar en contacto con ningún aparato reduciendo el riesgo de contaminación por radiación.

En caso de que sea posible aplicar esta tecnología al entorno deseado se desarrollará un cliente para un sistema de monitorización de plantas nucleares basado en el dispositivo de Microsoft Kinect para Windows. El usuario deberá navegar por el sistema a través de gestos corporales de manera natural. Este tipo de interfaces se conocen como Interfaces de Usuario Naturales (NUI en sus siglas inglesas).

El usuario podrá elegir entre cuatro opciones, a saber: Visualización de un panel general del estado de la planta, selección y visualización de esquemas y figuras, consulta de los valores disponibles en el sistema de monitorización, visualización de un gráfico y una estadística de las alarmas producidas en la planta.
El autor de este proyecto no conoce que exista actualmente ningún sistema con un cliente como este en uso en plantas nucleares tanto dentro como fuera de España.

Por tanto la finalidad del proyecto es crear un cliente basado en movimientos corporales y una respuesta gráfica a dichos movimientos que permita reducir el riesgo de contaminación por radiación en los operarios de la planta nuclear a la vez que éstos navegan por el sistema con movimientos naturales.
Debido a los requisitos del sistema de desarrollo de la tecnología Kinect el cliente será desarrollado en Windows Presentation Foundation (WPF) utilizando para ello los lenguajes XAML y C#, adoptando una metodología iterativa e incremental.

Palabras Clave

Kinect, Kinect for Windows, WPF, NUI, Nuclear, Monitorización de planta

Abstract

This Master Thesis aims to determine if Kinect technology has nuclear applications to develop a monitor system that can be used by nuclear power plant workers touching any device reducing radiation contamination risks.

In case it is possible to apply the technology to the desired nuclear environment a client for the monitor system will be developed based on Microsoft's device Kinect for Windows. The user should used the system through natural corporal movements. This kind of interfaces are called Natural User Interfaces (NUI).

The user should be able to choose between four different options: Visualization of a dashboard showing the state of the power plant, selection and visualization of different figures, queries asking for data values available on the monitor system and visualization of a graph and statistics of the last alarms on the power plant.
At this time there is no such client in use in any nuclear power plant neither in Spain nor outside.

This project looks to create a corporal movements based client giving a graphical response to those movements reducing radiation contamination risks to the power plant workers at the same time they use the system through natural movements.
The client will be developed using Windows Presentation Foundation (WPF) which uses XAML and C# languages. An iterative and incremental software development process is followed.

Keywords

Kinect, Kinect for Windows, WPF, NUI, Nuclear, Monitorización de planta

Índice General

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En este capítulo se describe una breve introducción al problema que pretende solventar este Proyecto de Fin de Carrera. Tras la descripción se expone la estructura del documento.

Índice

1.1. Introducción

Las plantas nucleares son una importante fuente de energía como demuestra el 21% de cobertura que tuvo la energía nuclear sobre la demanda anual de energía eléctrica peninsular en el año 2011 [REE 2012]. Sin embargo, dentro de una planta nuclear existen zonas contaminadas con radiación donde es necesario minimizar el riesgo de contaminación. En este sentido se toman multitud de medidas centradas en proteger a las personas de esta radiación.

Los equipos de las zonas contaminadas están expuestos durante largo tiempo a dicha radiación resultando en la contaminación del propio equipo. Utilizarlos de manera tradicional, es decir mediante el teclado y el ratón, aumenta el riesgo de contaminación de los operarios a la vez que supone un coste extra debido a que cuando se retira un equipo de la zona contaminada es preciso realizar una limpieza del mismo para asegurarse de que no hay restos de radiación.

Existe un dispositivo que permite realizar un seguimiento del usuario y de la posición de su cuerpo a la vez que aporta otras funcionalidades como reconocimiento de voz, una cámara RGB o un sensor de profundidad. Se trata del dispositivo Kinect [KNCT 2012] desarrollado por Microsoft.

El objetivo del presente Proyecto de Fin de Carrera es investigar la posible aplicación de la tecnología Kinect en este ambiente de manera que no sea necesario tocar los equipos para interactuar con ellos, creando así un ambiente de trabajo más seguro al eliminar un potencial foco de contaminación para los operarios.

Una vez analizadas las posibilidades de esta tecnología, en el caso de que sea aplicable a este entorno se creará un cliente para un sistema de monitorización de la planta nuclear que permita utilizar los equipos a través de gestos corporales naturales.

Este sistema se trata de IDbox [IDb 2012], un único sistema desarrollado por *CIC Consulting Informático* que integra, procesa y analiza toda la información de una planta. A través de IDbox se pueden consultar datos de todo tipo de sensores instalados en la planta a razón de incluso 10.000 muestras por segundo. No sólo se pueden consultar los datos sino que el sistema muestra análisis de los datos tanto históricos como en tiempo real.

Otra de las ventajas de este método de interacción es que al eliminar la necesidad de contacto con el equipo éste podría ser trasladado fuera de la zona contaminada de tal manera que los operarios interactuasen con el sistema a través de una gruesa placa de metacrilato que protegiese los equipos de la radiación evitando su posterior limpieza abaratando el mantenimiento de los mismos.

Para crear dicho cliente se utilizará el Kit de Desarrollo de Kinect para Windows [KfW 2012] dentro de una aplicación construida con tecnología .NET sobre Windows Presentation Foundation [NATHAN 2006].

1.2. Estructura del Documento

Tras este capítulo introductorio, el resto de la presente memoria se estructura tal como se describe a continuación:

Capítulo 2: Estado del Arte. Analiza la situación actual de la tecnología Kinect desde su origen como un dispositivo orientado a los videojuegos hasta el actual enfoque de desarrollo para aplicaciones de escritorio y aplicaciones empresariales.
Detalla también la creación de un proyecto de prueba de concepto para evaluar las posibilidades de esta tecnología en el entorno nuclear.

Capítulo 3: Planificación del Proyecto. Expone la planificación del cliente Kinect para IDbox. Describe el ámbito funcional del cliente, la metodología usada en su desarrollo, los requisitos de alto nivel del proyecto, las iteraciones que se han seguido en el desarrollo del cliente y las herramientas utilizadas.

Capítulo 4: Tecnología usada. Describe la tecnología usada en el desarrollo del cliente Kinect. En particular se da una descripción de Windows Presentation Foundation, del Kit de Desarrollo de Kinect para Windows y de los Servicios Web.

Capítulo 5: Sistema de reconocimiento de gestos. Detalla el proceso de desarrollo de la segunda iteración del proyecto del cliente Kinect para IDbox en la que se construye un reconocedor de gestos corporales. Se explican los objetivos de la iteración, la ingeniería de requisitos, su implementación y la verificación de su correcto funcionamiento.

Capítulo 2

Estado del Arte

En este apartado se redacta el análisis inicial realizado acerca del estado en el que se encuentra en la actualidad la tecnología Kinect [KNCT 2012] y su posible aplicación al entorno nuclear. Este estudio se desarrolló previamente al diseño del cliente Kinect para IDbox [IDb 2012], un sistema de monitorización de plantas que está presente en varias plantas nucleares.

La metodología empleada ha consistido en una primera etapa de investigación puramente teórica en la que se ha recabado información sobre la tecnología Kinect y se han analizado tanto tutoriales introductorios al desarrollo para Kinect como aplicaciones de código abierto e ideas de aplicaciones de código cerrado o de proyectos futuros.

Posteriormente, el estudio se ha centrado en una perspectiva más práctica. Se ha desarrollado un prototipo siguiendo los tutoriales encontrados que demuestra la capacidad de la tecnología y valora si es posible aplicarlo al ámbito deseado.

Índice

2.1. Dispositivo Kinect

Kinect es un dispositivo creado por Microsoft que originalmente se diseñó para la consola de videojuegos Xbox 360 [KX360 20120] con el propósito de que los jugadores pudiesen interactuar con la consola de una manera más natural e intuitiva de manera que la jugabilidad

fuese mucho más ágil y los juegos no requiriesen apenas de aprendizaje por parte de los usuarios para poder dominarlos.

Este dispositivo consta de tres partes diferenciadas. Una sistema de detección de profundidad a base de un sensor de rayos infrarrojos, una cámara de vídeo RGB y un array de cuatro micrófonos para reconocimiento de voz. Gracias al sistema de profundidad y a la cámara RGB el dispositivo es capaz de reconocer hasta seis jugadores y localizarlos en un espacio delante del dispositivo posicionándolos en un sistema de ejes X, Y, Z pudiendo además hacer un seguimiento del esqueleto de dos de los jugadores posicionando veinte puntos del esqueleto en el mismo sistema de ejes representando los puntos de la figura 2.1.

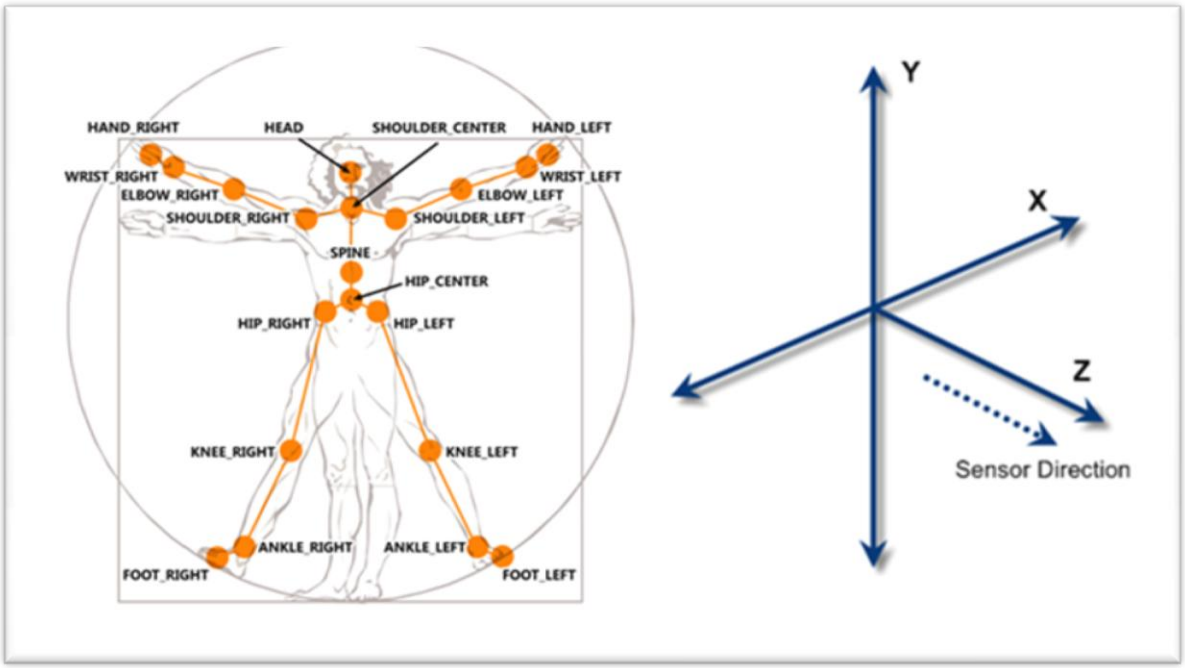

Figura 2.1: Puntos del esqueleto reconocidos por Kinect
Tomada de [DUMONT 2012]

El sensor de profundidad consiste en un proyecto de rayos infrarrojos los cuales son invisibles al ojo humano y no afectan a la usabilidad del dispositivo y un sensor CMOS monocromo que capta los rayos que rebotan en el entorno formando así una escena 3D del espacio de uso del dispositivo [KNCT 2012].

Gracias a esta tecnología el jugador puede interactuar con la consola sin necesidad de manipular ningún dispositivo de entrada pudiendo moverse libremente por el espacio de juego.

Esta tecnología pensada en un principio para ser usada en videojuegos rápidamente atrajo la atención de los desarrolladores al lanzarse el concurso *The Open Kinect Project* [OPK] por parte de la empresa *Adafruit Industries* en el que se instaba a crear un controlador de código abierto para usar el dispositivo Kinect sin necesidad de la consola Xbox 360.

Tras la finalización del concurso muchos desarrolladores comenzaron a crear aplicaciones para utilizar Kinect no como una herramienta para videojuegos sino como una interfaz de entrada para PCs consiguiendo por ejemplo controlar el ratón de un ordenador sobre Windows (Kinect used as a mouse! [KMC]) o una herramienta de dibujo basada en Kinect (Kinect Paint [KP]).

Viendo el gran interés de los desarrolladores Microsoft decidió lanzar un Kit de Desarrollo (SDK por sus siglas en inglés) para poder crear aplicaciones que usen el dispositivo en PCs de manera oficial dando soporte a los desarrolladores. La primera versión estable del SDK se lanzó el 1 de febrero del año 2012 junto con una versión del dispositivo diseñada específicamente para usarse en un PC sobre Windows, con nombre Kinect para Windows [KfW 2012]. Sin embargo esta versión del dispositivo no estuvo disponible en España hasta principios de abril del mismo año.

Este nuevo dispositivo mejora las características que aporta Kinect dando un mayor rango de resoluciones disponibles en la cámara RGB al igual que en el sensor de profundidad, una mejora sustancial del audio y el reconocimiento de voz, se ha mejorado la robustez de los sensores, entre ellos la estabilidad de los drivers y el tiempo de ejecución.
Como novedad este dispositivo además cuenta un "modo cercano" o "*near mode*" en inglés que permite detectar usuarios desde una distancia más cercana al sensor. Mientras que Kinect para Xbox 360 reconoce usuarios desde los 0'4 metros a los 4 metros de distancia reconociendo el esqueleto únicamente desde los 0'8 metros a los 4 metros, con el nuevo *near mode* se pueden reconocer esqueletos desde los 0'4 metros de distancia a los 3 metros de distancia del sensor [KFD].
Se puede ver una comparativa entre el reconocimiento de ambos modos en la figura 2.2.

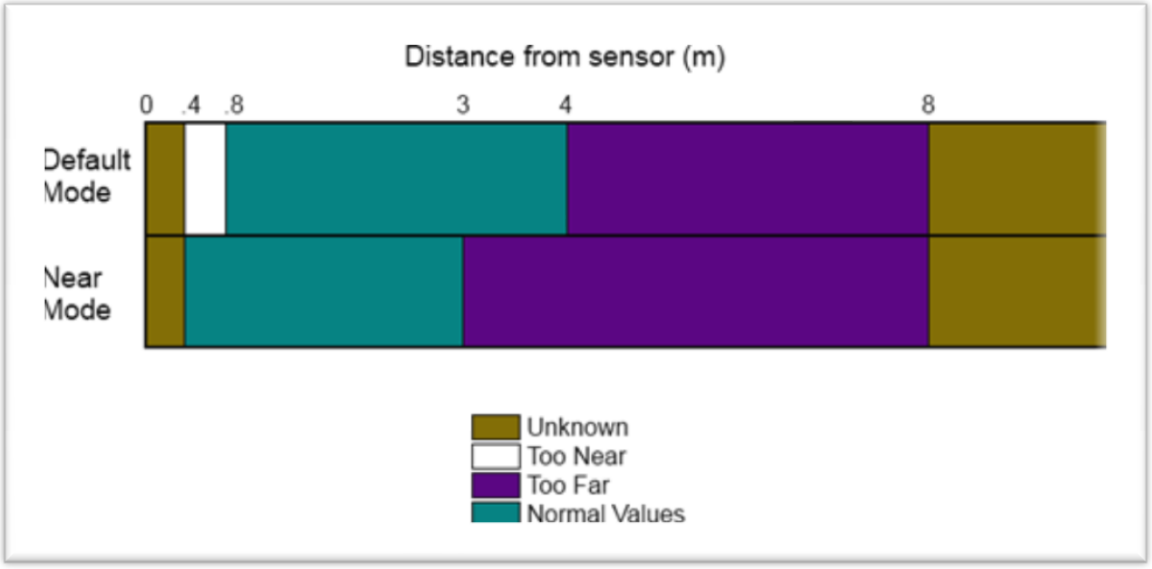

Figura 2.2: Distancias para el reconocimiento del esqueleto en Kinect
Tomada de [KBNM 2012]

Tras la primera versión estable del SDK se lanzó la versión 1.5 el 21 de mayo de 2012. Las innovaciones más destacables de esta versión son:

- Reconocimiento facial: Un SDK que permite reconocer y posicionar distintas partes de la cara haciendo un seguimiento de las mismas similar al que ya se hacía con el esqueleto.
- Modo sentado: Se reconoce a las personas sentadas reconociendo su esqueleto y haciendo un seguimiento de 10 articulaciones.
- El seguimiento del esqueleto está soportado en el modo cercano o *near mode*. Hasta ahora sólo se reconocía al jugador pero no se hacía un seguimiento de cada una de sus articulaciones.
- Se ha introducido un dato nuevo en el seguimiento del esqueleto referente a la orientación de cada articulación.
- Mejoras en el rendimiento y en la calidad de los datos.

2.2. Tutoriales y Proyectos de Código Abierto

Para aprender a desarrollar para Kinect para Windows se han seguido una serie de tutoriales producidos por Microsoft que se lanzaron a la vez que la primera versión del SDK. Estos tutoriales forman las Kinect for Windows Quickstart Series en Channel 9 [KQSS].

De especial interés son los primeros tutoriales [KQS1] [KQS2] que enseñan a instalar correctamente el dispositivo y las herramientas necesarias que son el SDK y un conjunto de herramientas "Kinect for Windows Developer Toolkit" que permite observar diversos ejemplos de aplicaciones sencillas que utilizan cada aspecto de Kinect, pudiendo instalar el ejemplo para ver en detalle su código fuente.

El tercer tutorial [KQS3] enseña a hacer funcionar la cámara RGB y mostrar una imagen del usuario. Este tutorial está desfasado ya que una parte de él se centra en utilizar el elemento ColorImageViewer que traía la versión 1.0 del Toolkit el cual ya no podemos encontrar en versiones posteriores de dicho Toolkit por lo que hay que programar la transformación de la información que nos brinda la cámara RGB a los píxeles de la imagen que queremos mostrar.

El cuarto tutorial [KQS4] enseña cómo utilizar y mostrar la información que nos da el sensor de profundidad. Al igual que el tercer tutorial, este se encuentra desfasado ya que una parte de él enseña a utilizar el elemento DepthViewer que venía con el Toolkit 1.0 pero que en sus versiones más recientes no se encuentra disponible.

El quinto tutorial [KQS5] enseña a usar el seguimiento de esqueletos. Al igual que en los dos anteriores tutoriales, éste se encuentra desfasado ya que en la última parte de este tutorial nos muestran cómo utilizar el elemento SkeletonViewer que traía el Toolkit 1.0 pero que en las versiones más recientes no está disponible.

El sexto tutorial [KQS6] nos enseña a utilizar el reconocimiento de voz. Sin embargo este tutorial no es objeto de nuestro estudio ya que en nuestro entorno el equipo se puede encontrar separado de los usuarios por una placa de metacrilato de diez centímetros lo cual aislaría todo el sonido que el usuario pudiese emitir.

Para entender en mayor profundidad el desarrollo de aplicaciones para Kinect para Windows se han analizado algunos proyectos de código abierto alojados en la página Codeplex [CDP] como el proyecto Kinect Mouse Cursor [KMC] el cual permite utilizar el dispositivo para controlar el ratón en Windows a través del SDK ofrecido por Microsoft.

Del proyecto Kinect Mouse Cursor se ha aprendido a utilizar la librería dinámica user32 para simular eventos del ratón y se ha investigado después dicha librería viendo que también es capaz de simular eventos del teclado.

2.3. Aplicación básica de Kinect para Windows

Una vez analizada la tecnología Kinect se ha procedido a construir una aplicación básica de Kienct para Windows siguiendo la metodología de desarrollo en cascada.

La metodología en cascada es un proceso de desarrollo en el que los pasos de desarrollo son vistos hacia abajo como en una cascada de agua a través de las fases de análisis de requerimientos, diseño, implementación, verificación o pruebas y mantenimiento. Por este motivo es ideal para la creación rápida de un prototipo sencillo de prueba de concepto ya que no será necesario pulir el desarrollo de dicho prototipo mediante iteraciones o revisiones del proyecto.

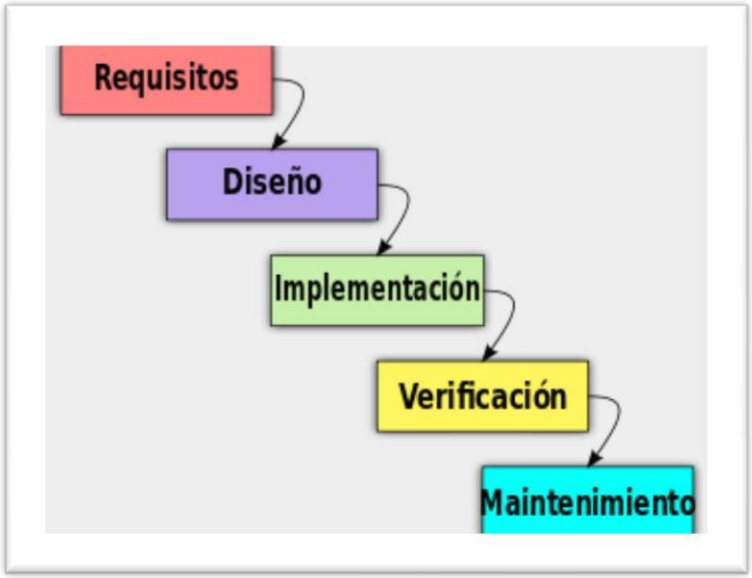

Figura 2.3: Modelo en cascada
Tomada de [WMC 2012]

En nuestro caso la última fase del proyecto, el mantenimiento, no se realiza ya que es un proyecto de prueba de concepto para verificar si es posible la aplicación de la tecnología Kinect en el entorno deseado.

2.3.1. Requisitos de la aplicación:

Los requisitos de la aplicación deseada son los siguientes:

- El usuario debe ser mostrado en una ventana en la aplicación.
- El esqueleto del usuario debe ser mostrado sobre la imagen del propio usuario una vez el dispositivo lo ha reconocido.
- El usuario debe ser capaz de adoptar distintas posiciones que funcionen como mecanismo de control de la aplicación.
- A través de dichas posiciones el usuario debe poder realizar distintas acciones:
  - Abrir el navegador por defecto.
  - Abrir la calculadora.
  - Pasar de una aplicación activa a otra.
  - Cerrar la aplicación activa.

2.3.2. Diseño:

El sistema se divide en tres subsistemas encargados cada uno de ellos de una parte del proyecto.

Estos subsistemas son la clase principal, el reconocedor de poses y el simulador de eventos de teclado.

2.3.2.1. Clase principal:

La clase principal tiene varias funciones siendo ésta la parte más importante del proyecto.

Es la encargada de recibir poner el marcha el sensor Kinect para Windows y recibir los eventos del sensor que nos proporcionarán la información disponible además de procesar dicha información. Tras procesar la información del sensor la clase principal muestra una imagen del usuario y el esqueleto del mismo usuario sobre dicha imagen.

La clase principal gestiona el control de la aplicación según la posición que adopte el usuario dejando al reconocedor de posiciones el trabajo de determinar la pose adoptada en cada momento.

Si se requiere efectuar un evento de teclado se gestiona a través del simulador de eventos de teclado.

2.3.2.2. Reconocedor de poses:

Tomando la información disponible sobre el esqueleto en un preciso momento el reconocedor de poses es capaz de determinar si el usuario está adoptando una de las poses previamente determinadas para el control de la aplicación.

2.3.2.3. Simulador de eventos de teclado:

El simulador de eventos de teclado utiliza la librería dinámica user32 para mandar eventos de teclado al sistema operativo pudiendo así simular la pulsación de una tecla y el levantamiento de dicha tecla.

2.3.3. Implementación:

Para la implementación del proyecto de prueba de concepto se utiliza el Kit de Desarrollo de Kinect para Windows proporcionado por Microsoft sobre tecnología .NET utilizando Windows Presentation Foundation (WPF) que se basa en lenguaje declarativo XAML para la interfaz de interacción y lenguaje C# para la lógica del negocio, propiciando una arquitectura Modelo Vista Controlador.

El proyecto se ha implementado mediante la herramienta Visual Studio 2010 que proporciona ayudas al programador como un diseñador visual de interfaces con el que prácticamente no es necesario programar en lenguaje XAML ya que mediante un sistema Drag and Drop se construye la interfaz de interacción produciendo el código necesario para la ejecución de la aplicación.

Se ha dividido el proyecto en tres clases correspondiéndose con los subsistemas planteados en el diseño del proyecto:

- La clase MainWindow implementa la clase principal de la aplicación.
- La clase Poses implementa el reconocedor de poses.
- La clase Keys implementa el simulador de eventos de teclado.

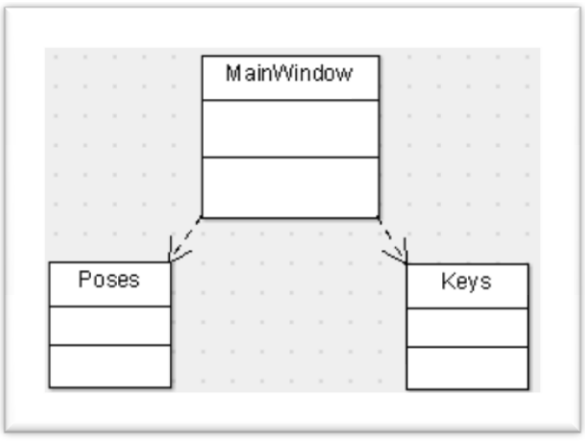

Figura 2.4: Diagrama de clases del prototipo

A continuación se describe la implementación de cada una de las clases por separado.

2.3.3.1. Poses:

Esta clase reconoce la pose que está adoptando el usuario a través de la información de su esqueleto disponible en el objeto Skeleton que toma como entrada su método principal.

El método princpal de esta clase toma como argumento de entrada un objeto de la clase Skeleton que contiene información relativa al esqueleto del usuario y comprueba las distintas poses que se han determinado en varios métodos de la misma clase.

Los métodos que comprueban las distintas posiciones analizan los datos del esqueleto y cotejan si se encuentran en la posición adecuada para adoptar una pose. Un ejemplo de esta pose es levantar el mano derecha.

Para analizar si el usuario ha levantado su mano derecha se comprueba la posición de la mano relativa a la posición del hombro derecho, el codo derecho y la muñeca derecha.

Como cada persona tiene unas dimensiones diferentes no es recomendable utilizar medidas absolutas en la detección de posiciones o gestos, por lo que se toma una distancia de referencia que en nuestra caso es la distancia vertical entre el centro del cuerpo y el centro de la cadera tomada mediante los objetos Joint del esqueleto que representan las articulaciones, en concreto los objetos Spine y HipCenter.

Una vez se dispone de la distancia de referencia se comprueba si la mano derecha se encuenta a una posición tal que:

- En el eje vertical sea mayor que la posición del hombro derecho más tres veces la distancia de referencia. A la vez debe ser mayor que la posición de la muñeca derecha más un cuarto de la distancia de referencia.
- En el eje horizontal se encuentre entre la posición del codo derecho menos la distancia de referencia y la posición del codo derecho más dos veces la posición de referencia.
- En el eje Z que representa la profundidad se encuentre entre la posición del codo derecho menos cinco veces la distancia de referencia y la posición del codo derecho.

2.3.3.2. Keys:

Esta clase simula eventos de teclado gracias a la librería dinámica user32.

A través de esa librería se accede al método keybd_event que permite tanto oprimir como soltar una tecla que está representado mediante un valor en forma de Byte.

Las teclas utilizadas están codificadas de la siguiente manera:

Tabla Alt: VK_MENU = 0x12
Tecla Tab: VK_TAB = 0x09
Tecla F4: VK_F4 = 0x73

Esta clase contiene varios métodos que conforman las secuencias de eventos de teclado requeridas por la aplicación.

2.3.3.3. MainWindow:

Esta clase dispone de varios métodos en los que se realizan las siguientes funciones:

Constructor MainWindow:

Inicializa los componentes de la interfaz de usuario mediante un método de creación automática que proporciona Visual Studio 2010.

Window_Loaded:

Este método se ejecuta en el momento de carga de la ventana de la aplicación. Es el encargado de detectar el sensor Kinect para Windows y de activar las distintas características del sensor además de asignar un método al evento AllFramesReady del sensor que se activará cada vez que el sensor tenga todos los datos disponibles.

_sensor_AllFramesReady:

Este método se activa cada vez que el sensor tenga todos los datos disponibles y se encarga de mostrar en la ventana principal de programa la imagen del usuario y el esqueleto de dicho usuario en la misma imagen.

Para ello se toman los datos de la cámara RGB en forma de bytes y se copian a un array de bytes mediante el método CopyPixelDataTo proporcionado por el SDK de Kinect para Windows.

Una vez tomados los datos se escriben en una imagen de mapa de bits mediante el método WritePixels propio del elemento WriteableBitmap.

Posteriormente se muestra el esqueleto del usuario dibujando sobre la misma imagen de mapa de bits una serie de puntos y rectas representando los datos disponibles del seguidor de esqueletos, obtenidos a través de diversos métodos del SDK.

Además esta clase se encarga del control de la aplicación solicitando al reconocedor de poses la pose que está adoptando el usuario y actuando en consecuencia. Para ello toma los datos del primer esqueleto disponible reconocido por el sensor.

Si el usuario levanta la mano izquierda se abrirá la calculadora.

Si el usuario levanta la mano derecha se abrirá el navegador.

Si el usuario estira su brazo izquierdo hacia la izquierda se cerrará la aplicación actual mediante la simulación de las teclas Alt y F4 simultáneamente.

Si el usuario estira su brazo derecho hacia la derecha se cambiará de aplicación mediante la simulación de las teclas Alt y Tab simultáneamente.

2.3.4. Verificación:

La verificación del proyecto se ha hecho mediante la interacción de un usuario con la aplicación adoptando todas las poses disponibles y comprobando el correcto funcionamiento de la aplicación.

2.4. Sumario

En este capítulo se ha explicado en qué consiste el dispositivo Kinect así como el entorno de aprendizaje de su uso. También se ha descrito el desarrollo de la aplicación de prueba de concepto que investiga si el dispositivo Kinect es aplicable al entorno nuclear deseado.

Capítulo 3

Planificación del Proyecto

En este capítulo se describe la planificación que tendrá la creación del cliente basado en la tecnología Kinect para el sistema de monitorización de plantas IDbox. Se muestra en líneas generales el ámbito funcional del cliente, la metodología de desarrollo utilizada, los requisitos de alto nivel, las iteraciones programadas, las herramientas utilizadas para su desarrollo y la construcción de prototipos.

Índice

3.1. Ámbito Funcional del Proyecto

En esta sección se describe el ámbito funcional del cliente Kinect para IDbox [IDb 2012]. Se trata de un cliente basado en una interacción gestual sin contacto entre el usuario y el equipo.

IDbox es un sistema desarrollado por *CIC Consulting Informático* que integra, procesa y analiza toda la información de una planta. A través de IDbox se pueden consultar datos de todo tipo de sensores instalados en la planta a razón de incluso 10.000 muestras por segundo. No sólo se pueden consultar los datos sino que el sistema muestra análisis de los datos tanto históricos como en tiempo real.

Aunque IDbox es una solución creada para cualquier tipo de planta, ya sea eólica, nuclear o una red eléctrica por ejemplo, el objetivo del cliente basado en Kinect es su integración en centrales nucleares.

Las centrales nucleares tienen zonas con un alto riesgo de contaminación por radiación debido a los materiales que manejan, en las que los operarios deben pasar un chequeo al entrar y salir de la sala para comprobar los niveles de radiación de su cuerpo. Gracias a las rigurosas medidas de seguridad tomadas por parte de las centrales, es posible trabajar en estas zonas contaminadas sin que sea dañino para la salud. Sin embargo en estas zonas hay equipos permanentes que absorben esta contaminación constantemente y suponen un riesgo potencial para los trabajadores.

Como medida para reducir este riesgo de contaminación surge el actual proyecto que pretende eliminar el contacto con los equipos del sistema de monitorización permitiendo a los operarios consultar datos de la central sin necesidad de tocar nada. Incluso mediante este cliente los equipos se pueden encontrar alojados en una sala contigua a la zona contaminada evitando así tener que limpiar la radiación absorbida por estos equipos cuando se retiren de la central, reduciendo así también los costes de mantenimiento de los mismos.

El cliente permitirá a los trabajadores consultar datos de la central ofreciendo las siguientes funciones:

1. El cliente ofrecerá en menú principal con cuatro opciones: Indicadores, Sinópticos, Alarmas y Análisis.
2. Dentro de la opción Indicadores se mostrará un cuadro de mandos con la situación general actual de la planta en el que se podrán observar indicadores en forma de semáforo sobre el estado de la central. Así mismo se mostrarán contadores sobre el combustible y los días de actividad de la planta. Por último se mostrarán variables del entorno.
3. El usuario podrá seleccionar cada uno de estos datos para acceder a los detalles de los mismos, en forma de tabla o de gráfico dependiendo del dato seleccionado.
4. Accediendo a la opción de Sinópticos se mostrará un selector de imágenes representativas de la central y sus elementos.
5. El usuario será capaz de seleccionar la imagen deseada para verla en mayor tamaño pudiendo dentro de este visor hacer zoom sobre la imagen y desplazarse por ella.
6. Dentro de la opción Alarmas se mostrará un gráfico y una tabla estadística con las últimas alarmas producidas en la central.
7. Accediendo a la opción Análisis se ofrecerá al usuario un menú donde podrá seleccionar la categoría del dato que quiere consultar.
8. Una vez seleccionada la categoría se mostrarán al usuario los datos agrupados en dicha categoría permitiéndole seleccionar un dato para consultar su valor en tiempo real, una tabla histórica del dato o una gráfica que represente el histórico de dicho dato.

Tras describir a grandes rasgos el funcionamiento del cliente, la siguiente sección proporciona una visión de la metodología que será utilizada para su desarrollo, así como las justificaciones para la elección de la misma.

3.2. Metodología de Desarrollo

Esta sección muestra en detalle la metodología utilizada para el desarrollo del cliente Kinect y los motivos para el uso de dicha metodología.

La metodología de desarrollo que se ha seguido es el Modelo Iterativo e Incremental [BitSpe 2006]. Se trata de un modelo evolutivo basado en el desarrollo de programas de manera incremental, permitiendo al desarrollador sacar ventaja de lo aprendido a lo largo de las anteriores etapas del proyecto, incrementando versiones entregables del sistema que con cada iteración le otorgan más funcionalidades.

El uso del modelo iterativo a incremental requiere de una clara definición del problema que se busca solucionar y del sistema a desarrollar. Con esta idea se crea la primera versión del proyecto en la primera iteración. Esta es la base del proyecto que cuenta con las funcionalidades más básicas.

Sobre esta base se ejecutan una serie de incrementos añadiendo en cada uno de ellos funcionalidades adicionales a la aplicación cumpliendo con una serie de requisitos específicos de cada iteración. Las primeras iteraciones implementarán las funcionalidades más básicas de manera que se puede construir el sistema de manera incremental añadiendo funcionalidades secundarias o más complejas en las siguientes iteraciones. De esta manera con el añadido de nuevas funcionalidades se irán cumpliendo los requisitos totales del sistema también de manera incremental.

A cada iteración dentro del desarrollo se le aplican las disciplinas fundamentales del desarrollo de software disponiendo ésta de las siguientes fases: Análisis de requisitos, diseño, implementación y verificación de su funcionamiento. La primera iteración da como resultado la versión inicial de la aplicación aportando una funcionalidad muy básica y cumpliendo con los requisitos más básicos del sistema. Cada iteración aporta más funcionalidades al sistema y cumple con más requisitos de manera que en la última iteración se obtiene el producto final que cumple todos los requisitos planteados para el sistema.

Esta metodología se adapta perfectamente al desarrollo del cliente Kinect para IDbox ya que es un proyecto experimental en el que poder contar con un prototipo al final de cada iteración sirve para saber si realmente el proyecto se adapta a los objetivos y corregir aquellos puntos que no satisfagan los deseos hacia el producto final que se busca.

Gracias a que se trata de un proceso de desarrollo evolutivo es relativamente sencillo incorporar cambios en los requisitos del sistema, lo cual es muy común en proyectos de este tipo.

3.3. Requisitos de Alto Nivel del Cliente

Esta sección contiene los requisitos de alto nivel del cliente Kinect para IDbox que habrán de verse satisfechos con el producto final del desarrollo.

Los requisitos del cliente Kinect son los siguientes:

Referencia	Requisito
R01	En la ventana del cliente se mostrará una imagen del usuario y su esqueleto
R02	En todo momento se indicará si se ha reconocido algún gesto
R03	Se dispondrá de un menú mediante el que se accede a las opciones principales
R04	El usuario podrá navegar a través del cliente mediante gestos corporales
R05	Se dispondrá de un visor de sinópticos
R06	El usuario podrá seleccionar el sinóptico que quiera visualizar
R07	El usuario podrá navegar por del sinóptico mediante el uso de zoom y desplazamientos
R08	Será posible acceder a un cuadro de mando con el estado general de la central
R09	El usuario podrá ver detalles del estado de la central
R10	El usuario podrá realizar consultas sobre los valores de los sensores de la central
R11	Se podrá acceder a datos históricos de los sensores de la central en forma de tabla o de gráfica
R12	Se dispondrá de una tabla con las últimas alarmas producidas en la central
R13	Se dispondrá de un gráfico con las últimas alarmas producidas en la central

Tabla 3.1: Requisitos del cliente

3.4. Iteraciones

En esta sección se muestran las iteraciones planificadas para el desarrollo del cliente Kinect para IDbox a partir de los requisitos enumerados en la sección 3.3.

Para desarrollar el cliente se ha partido del prototipo de prueba de concepto realizado en la fase de investigación de este proyecto, aprovechando el proyecto WPF ya creado y el reconocedor de poses. Este proyecto ya cumple con algunos de los requisitos del cliente, en concreto los requisitos R01 y R02, además de una versión preliminar del requisito R03 ya que la navegación no se efectúa a través de gestos sino de poses, esto será cambiado en las próximas iteraciones.

Las iteraciones planificadas son las siguientes:

1. Menú navegable por el usuario. (Ver R03)
2. Sistema de reconocimiento de gestos. (Ver R04)
3. Asignación de gestos para la navegación por el menú.
4. Visor de sinópticos. Creación del sistema de zoom para el visor. (Ver R05, R06 y R07)
5. Ambientación gráfica del cliente.
6. Cuadro de mandos. (Ver R08)
7. Selector de categoría de datos para las consultas. (Ver R10)
8. Sistema de reconocimiento del usuario.
9. Sistema de selección de datos para el cuadro de mandos y las consultas. Menú de selección de las consultas. (Ver R09 y R11)
10. Sistema de movimiento de los sinópticos. Gráfico y tabla de alarmas producidas en la central. (Ver R07, R12 y R13)

3.5. Herramientas utilizadas para el Desarrollo del Cliente

En esta sección se describen las herramientas utilizadas para la creación del cliente.

El proyecto ha sido desarrollado con la ayuda del entorno de desarrollo Visual Studio 2010 [RaGaAnMi 2010]. Dentro del proyecto en Visual Studio 2010 se han hecho referencias para poder utilizar servicios web [RIBAS 2010], los cuales fueron probados a través del navegador Google Chrome [CHROME 2012].

Para la modificación de las imágenes vectoriales que se utilizaron en primer lugar en el visor de sinópticos se utilizó la herramienta Notepadd++ [N++ 2011].

Las imágenes que forman parte de la interfaz de usuario han sido desarrolladas por la empresa CIC Consulting Informático. Sin embargo algunas de ellas han requerido alguna modificación. Para ello se ha utilizado el programa GIMP [MuTri 2007].

El desarrollo del cliente ha sido realizado sobre el sistema operativo Microsoft Windows 7 [SiStBo 2010].

3.6. Sumario

Durante este capítulo se ha descrito la planificación del proyecto del cliente Kinect para IDbox, definiendo el ambiente funcional en el que se encuentra. También se ha indicado la metodología seguida para el desarrollo del cliente, así como los requisitos de alto nivel del proyecto y las iteraciones que se han seguido en su desarrollo.

Capítulo 4

Tecnología usada

En este capítulo se describe brevemente la tecnología usada en el desarrollo del cliente Kinect para IDbox. Se explica la tecnología Windows Presentation Foundation además del Kit de Desarrollo de Kinect para Windows y el uso de Servicios Web.

Índice

4.1. Windows Presentation Foundation

En esta sección se describe la tecnología Windows Presentation Foundation [NATHAN 2006] (WPF) de Microsoft.

Se trata de un marco de interfaz de usuario de nueva generación para crear aplicaciones cliente enriquecidas e interactivas. Es un subconjunto de .NET Framework [ROEBUCK 2011], es el subsistema de Windows orientado a unificar los mecanismos de creación y gestión de interfaces de usuario.

WPF utiliza un lenguaje de marcado basado en XML llamado XAML [MOORE 2010]. Este lenguaje declarativo es usado para implementar la interfaz de una aplicación pudiendo crear ventanas, cuadros de diálogo, páginas y controles de usuario, así como rellenarlos con controles, formas y gráficos.

Esta tecnología se complementa con la lógica de negocio de la aplicación codificada en la plataforma .NET sobre el lenguaje C# o Visual Basic.

4.2. Kinect SDK

Esta sección habla del Kit de Desarrollo (SDK por sus siglas en inglés) [MILES 2012] de Kinect para Windows publicado por Microsoft el 1 de febrero del año 2012.

El SDK de Kinect para Windows da soporte a aplicaciones creadas en C++, C# o Visual Basic a través de Visual Studio 2010. Aporta unas librerías que referenciadas en el proyecto permiten acceder al sensor Kinect y obtener la información que éste aporta.

Gracias a estas librerías se puede controlar el sensor pudiendo activar o desactivar sus características, inclinándolo hacia arribo o hacia abajo gracias a los motores de los que dispone, activando los distintos modos del sensor (modo normal o sentado), conocer el estado del sensor y acceder a muchas más funcionalidades.

4.3. Web Services

En esta sección se explica lo que es un servicio web o web service [RIBAS 2010] en inglés.

Un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones a través de la red, ya sea por internet o en la intranet de una empresa.

Los servicios web utilizados en el desarrollo del cliente se incluyen en el proyecto como una referencia pudiendo así acceder a los métodos que ofrecen para obtener datos.

En Visual Studio 2010 se deben incluir como una referencia web y crear un objeto del cliente del servicio invocando un constructor específico del servicio web. Una vez creado se puede extraer del objeto cliente la información que puede aportar dicho servicio web usando sus métodos públicos.

4.4. Sumario

Este capítulo ha descrito la tecnología usada en el cliente Kinect para IDbox, explicando la tecnología WPF, el Kit de Desarrollo de Kinect para Windows y los servicios web.

Capítulo 5

Sistema de reconocimiento de gestos

Este capítulo trata sobre la segunda iteración del desarrollo del cliente Kinect para IDbox. En esta iteración se crea el sistema de reconocimiento de gestos corporales y se detallan los gestos a utilizar para el control del cliente.

Índice

5.1. Objetivos

En esta sección se detallan los objetivos de esta iteración del proyecto.

A pesar de que el dispositivo Kinect ofrece información sobre la posición del esqueleto del usuario en cada momento al igual que de las articulaciones del mismo, no tiene implementado ningún sistema para reconocer gestos preestablecidos ni ningún tipo de almacenaje de los datos del esqueleto.

La segunda iteración del proyecto del cliente Kinect tiene como objetivo construir el sistema de reconocimiento de gestos corporales que permitirá determinar los gestos que realice el usuario para controlar la aplicación.

5.2. Ingeniería de Requisitos

En esta sección se muestran los casos de uso y el refinamiento de requisitos.

5.2.1. Casos de uso:

Los casos de uso de la segunda iteración corresponden al reconocedor de gestos del cliente y a los gestos predefinidos para el uso de la aplicación.

Un usuario debe poder controlar la aplicación sin necesidad de contacto con dispositivo alguno, empleando únicamente determinados gestos corporales que han sido designados a tal fin. El usuario utilizará los gestos definidos en la sección de diseño de este capítulo tanto para navegar por los menús como para seleccionar un elemento, navegar por los sinópticos o subir un nivel en el control de la aplicación.

5.2.2. Refinamiento de los requisitos:

Como consecuencia del análisis y diseño de esta iteración se han realizado los siguientes refinamientos en los requisitos del proyecto.

Referencia	Requisito
R04.1	El usuario podrá navegar a través de los menús con tres movimientos: Swype (izquierda o derecha), selección y volver
R04.2	Si el usuario mantiene la posición al final del movimiento de swype, el menú seguirá avanzando posiciones hasta que el usuario abandone dicha posición
R04.3	La navegación por los sinópticos se basará en el movimiento de zoom (hacia dentro o hacia fuera) y el de movimiento del sinóptico

Tabla 5.1: Refinamiento de requisitos. Segunda iteración

5.3. Implementación

En esta sección se describe la implementación del reconocedor de gestos así como de los gestos predefinidos para el control de la aplicación.

El primer problema que se ha hecho claro en la implementación de un reconocedor gestual es la falta de información histórica sobre la posición de las articulaciones del esqueleto, ya que es posible acceder a las posiciones actuales a través del dispositivo Kinect pero no a las posiciones pasadas.

Para poder contar con estos datos históricos del esqueleto es necesario almacenarlos en una lista dentro del reconocedor de gestos e ir actualizando esta lista cada vez que se disponga de nueva información sobre el esqueleto.

Sin embargo esto no es suficiente ya que para controlar el proceso del gesto queremos saber el tiempo en el que se detectó cada posición del esqueleto, dado que un gesto hecho extremadamente lento no es válido para el control de la aplicación. Aunque el dispositivo Kinect tenga una tasa de actualización de 30 frames por segundo, dicha tasa no es totalmente precisa. Por este motivo no sirve con sólo hacer una lista de las últimas informaciones sobre el esqueleto sino que hay que crear una clase que almacene los datos tomados de un esqueleto y una marca de tiempo que indique el momento exacto en el que fueron tomados esos datos.

Para actualizar la lista, se guardan las últimas noventa informaciones que se tengan sobre el esqueleto, de modo que aproximadamente contamos con tres segundos de información sobre las posiciones que tomó el esqueleto y cada articulación del usuario. Con estos datos es posible comparar las posiciones de las articulaciones del usuario a través del tiempo y comprobar si se ha realizado algún gesto o no.

Los gestos se han definido de manera similar a como se definieron las posiciones en el capítulo 2.

En el desarrollo del cliente se han definido siete gestos que el usuario podrá utilizar. A continuación se describen los gestos predefinidos.

5.3.1. Swype izquierdo:

Este gesto es utilizado para avanzar en los elementos de un menú hacia la izquierda o hacia arriba. Consiste en, empezando con la mano izquierda más a la izquierda que la cadera, pasar la mano hacia la derecha hasta pasar la parte derecha de la cadera. Se puede ver una secuencia representando este gesto en la siguiente figura.

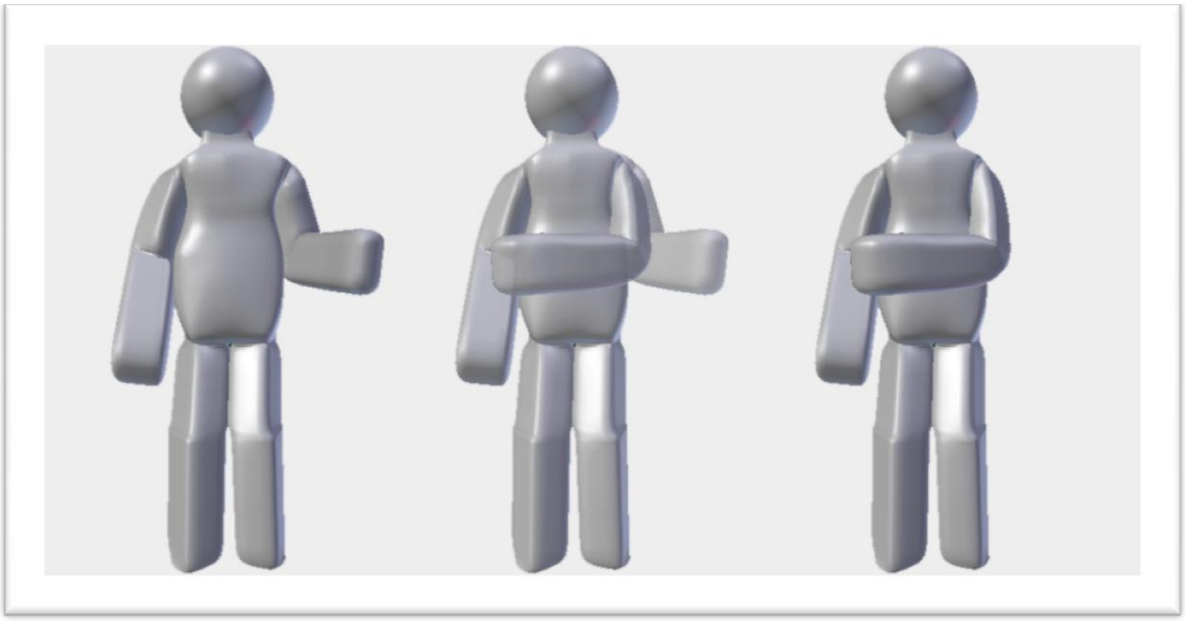

Figura 5.1: Gesto de swype izquierdo

5.3.2. Swype derecho:

El gesto de swype derecho se utiliza para avanzar en los elementos de un menú hacia la derecha o hacia abajo. Tiene la misma definición que el izquierdo con la salvedad de que este se realiza con la mano derecha, desde la derecha de la cadera hasta pasar la parte izquierda de la misma.

5.3.3. Mano izquierda arriba:

Este gesto es utilizado para seleccionar un elemento de un menú o para acceder al detalle de un elemento tanto del cuadro de mandos como del resultado de la consulta de datos.

El gesto de la mano izquierda arriba se realiza posicionando la mano izquierda por encima de los hombros del usuario formando un ángulo de 90º con el brazo y el antebrazo. La siguiente figura representa dicha posición.

Figura 5.2: Gesto de la mano izquierda arriba

5.3.4. Mano derecha arriba:

Este gesto es utilizado para subir un nivel en el control de la aplicación, es decir, si el usuario ha seleccionado una de las opciones del menú principal, usaría este gesto para regresar a dicho menú y así funcionaría también con elementos más internos de la aplicación.

Como ocurría con los gestos de swype, este gesto se define igual que el gesto de la mano izquierda arriba salvo que se realiza con el brazo derecho.

5.3.5. Ambas manos al frente:

El gesto de las dos manos al frente se utiliza para controlar el zoom cuando se ha seleccionado un sinóptico y se está visualizando dicho sinóptico.

Para realizar este gesto se han de estirar ambos brazos hacia el frente de manera que queden totalmente estirados. Una vez en esta posición tomamos el control del zoom y podemos ampliarlo separando las manos o reducirlo juntándolas.

5.3.6. Mano izquierda al frente:

Este gesto se utiliza para tomar el control del movimiento de un sinóptico una vez se ha seleccionado y se está visualizando dicho sinóptico.

Para realizar este gesto se ha de estirar el brazo izquierdo hacia delante de manera que quede totalmente estirado. Una vez en esta posición se toma el control del movimiento del sinóptico y al mover la mano manteniendo el brazo estirado hacia el frente, se mueve el sinóptico.

5.3.7. Mano derecha al frente:

Al igual que el gesto anterior, este gesto se utiliza para tomar el control del movimiento de un sinóptico una vez se ha seleccionado y se está visualizando dicho sinóptico.

Este gesto es igual que el gesto de la mano izquierda al frente salvo que se efectúa con la mano derecha. Una vez que tenemos el brazo derecho completamente estirado hacia delante tomamos el control del movimiento del sinóptico y podemos moverlo al mover la mano derecha.

5.4. Verificación

En esta sección se detalla la verificación realizada para comprobar el correcto funcionamiento de la aplicación tras la segunda iteración del proceso de desarrollo.

Habiendo terminado la fase de implementación se han realizado unas pruebas sobre la aplicación para comprobar el correcto funcionamiento de la misma y que se han cumplido los requisitos que se marcaron en esta iteración.

Con este objetivo se ha ejecutado el cliente y se han realizado los gestos predefinidos en la sección 5.3 comprobando que han sido reconocidos gracias al texto indicativo que se encuentra debajo de la imagen del usuario en la ventana del cliente.

5.5. Sumario

En este capítulo se ha descrito el proceso de desarrollo correspondiente a la segunda iteración del proyecto del cliente Kinect para IDbox en la que se ha creado el sistema reconocedor de gestos corporales. Este uno de los objetos más importantes del proyecto ya que en él se basa la interacción del usuario con el sistema.

Durante este capítulo se han expuesto los casos de uso de la segunda iteración, el refinamiento de requisitos adoptado tras el análisis y el diseño del reconocedor de gestos, la fase de implementación y la fase de verificación de la iteración donde comprobamos su correcto funcionamiento.

Capítulo 6

Sistema de reconocimiento del usuario

Este capítulo trata sobre la octava iteración del desarrollo del cliente Kinect para IDbox. En esta iteración se crea el sistema de reconocimiento del usuario de modo que ninguna persona que pase por el espacio de uso del cliente quite el control de la aplicación al usuario actual sin que este lo permita.

Índice

6.1. Objetivos

En esta sección se describen los objetivos que tiene esta iteración del proceso de desarrollo del proyecto.

Durante la fase de verificación de las iteraciones anterior del proyecto se ha detectado un problema a la hora de reconocer los gestos corporales que realizaba el usuario. Este problema consiste en que si otra persona entra en el espacio de uso del cliente y el dispositivo Kinect lo reconoce como un usuario y reconoce su esqueleto, ésta persona que tomaría accidentalmente el control de la aplicación haciendo que el usuario no fuese capaz de retomarlo hasta que sólo quedase él en el espacio de uso.

Debido a esto se requiere un sistema de reconocimiento del usuario que permita mantener el control de la aplicación a un usuario aunque otras personas pasen por delante del dispositivo Kinect.

6.2. Ingeniería de Requisitos

En esta sección se muestran los casos de uso y el refinamiento de requisitos.

6.2.1. Casos de uso:

Los casos de uso de la octava iteración corresponden al sistema de reconocimiento del usuario.

Un usuario debe poder controlar la aplicación y mantener el control sobre ésta en todo momento aunque alguna persona entre en el espacio de uso de la aplicación y el dispositivo Kinect realice el seguimiento de su esqueleto.

6.2.2. Refinamiento de los requisitos:

Como consecuencia del análisis y diseño de esta iteración se han añadido los requisitos al proyecto.

Referencia	Requisito
R14	Una vez se haya detectado un usuario sólo él tendrá el control del cliente
R14.1	El usuario deberá saludar a la pantalla para ser reconocido por el sistema
R14.2	Si el usuario desea liberar el control de la aplicación deberá saludar a la pantalla

Tabla 6.1: Refinamiento de requisitos. Octava iteración

6.3. Implementación

En esta sección se describe la implementación del sistema de reconocimiento del usuario.

Para poder implementar un sistema de reconocimiento del usuario lo primero es separar la información de los respectivos esqueletos de cada usuario ya que hasta esta iteración se había mantenido un único sistema de reconocimiento de gestos que almacenaba todos los datos de esqueletos que llegaban desde el dispositivo Kinect.

La información sobre los esqueletos llega a través del kit de desarrollo de Kinect para Windows en forma de un array de objetos Skeleton que contienen los datos de cada esqueleto. Contando con esto cada uno de los seis esqueletos que puede reconocer el dispositivo Kinect

tendrá su propio reconocedor de gestos de manera que no se mezclen los datos de varios esqueletos en un mismo registro histórico de los datos.

Para reconocer al usuario éste ha de realizar un gesto el cual consiste en saludar a la pantalla con la mano derecha, estando dicha mano por encima de los hombros.

Antes de proceder a la etapa de control de la aplicación se solicita a través de un método el esqueleto al que corresponde controlarla.

Dicho método comprueba si ya hay algún esqueleto que tenga el control de la aplicación buscando en los esqueletos disponibles si el ID de alguno de ellos coincide con el del esqueleto que tenía el control en la anterior iteración de la ejecución de la aplicación. Si es así, ese será el esqueleto que tenga el control, si no, se comprueba mediante el reconocedor de gestos de cada esqueleto si alguno ha realizado el gesto de toma de control.

Si no hay ningún esqueleto que tenga tomado el control y se detecta que alguno quiere tomarlo mediante el gesto correspondiente se actualiza la información, es decir, se identifica que un esqueleto tiene tomado el control mediante una variable global booleana, se asigna el objeto Skeleton a una variable global, igual que con el ID del esqueleto y con el reconocedor de gestos.

Gracias a los datos almacenados en variables sobre el esqueleto se puede ejecutar la fase de control de la aplicación mirando sólo datos de dicho esqueleto y en la próxima iteración comprobar si sigue estando disponible para seguir mirando sus datos o proceder a detectar un nuevo usuario.

Este sistema de reconocimiento de usuario también requiere una pequeña modificación en el control de la aplicación ya que si se detecta que el usuario saluda a la pantalla mientras tiene tomado el control ha de dejar de ser reconocido como el usuario actual. Esto se consigue reseteando las variables asignadas a la información del esqueleto actual.

6.4. Verificación

En esta sección se detallan las pruebas realizadas para la comprobación del correcto funcionamiento del sistema de reconocimiento de usuarios desarrollado en la octava iteración.

Para ello se ha ejecutado el cliente y un usuario ha entrado en el espacio de uso de la aplicación. Se ha comprobado que no se le detectaba como el usuario actual realizando los gestos predefinidos en el capítulo 5 y observando que no permitían la interacción con el cliente. En la siguiente figura se puede observar como el usuario no está reconocido como el usuario actual.

Figura 6.1: Usuario no reconocido por el sistema

A continuación el usuario realizó el gesto de toma de control saludando a la pantalla y prosiguió realizando el resto de gestos de la aplicación comprobando que esta vez sí le permitía controlarla.

Por último el usuario ha realizado el gesto de cese de control comprobando que de nuevo no se le reconocía como el usuario actual de la aplicación.

6.5. Sumario

En este capítulo se ha descrito la octava iteración del desarrollo del cliente Kinect para IDbox.

Durante este capítulo se ha hablado del proceso de desarrollo de este iteración en la que se ha construido el sistema de reconocimiento de usuarios del cliente detallando los objetivos de la iteración, los casos de uso del sistema de reconocimiento, los requisitos que ha sido necesario añadir debido al análisis y el diseño de la iteración, la implementación del sistema y la verificación que se ha realizado para comprobar su correcto funcionamiento.

Capítulo 7

Sistema de movimiento de los sinópticos. Gráfico y tabla de alarmas producidas en la central

Este capítulo trata sobre la décima y última iteración del desarrollo del cliente Kinect para IDbox. En esta iteración se crea el sistema de movimiento de los sinópticos así como la página que mostrará el gráfico y la tabla que representan las últimas alarmas producidas en la central.

Índice

7.1. Objetivos

En esta sección se describen los objetivos que tiene esta iteración del proceso de desarrollo del proyecto.

Los objetivos de esta última iteración se dividen en dos partes. Una de ellas se centra en la creación del sistema de movimiento de los sinópticos una vez que se ha seleccionado un sinóptico y la otra se centra en la creación de la página que muestra un gráfico y una tabla con los datos de las últimas alarmas producidas en la central nuclear.

7.2. Ingeniería de Requisitos

En esta sección se muestran los casos de uso y el refinamiento de requisitos.

7.2.1. Casos de uso:

Los casos de uso de la décima iteración corresponden al sistema de movimiento de los sinópticos y a la página donde se muestran el gráfico y la tabla con los datos de las últimas alarmas.

El usuario debe ser capaz de mover el sinóptico estirando uno de los dos brazos hacia el frente y moviendo la mano. De ese modo la imagen se moverá a la vez que el usuario posicione su mano en uno u otro sitio.

Cuando el usuario seleccione la opción Alarmas se mostrará un gráfico y una tabla con los datos de las últimas alarmas producidas en la central.

7.2.2. Refinamiento de los requisitos:

Como consecuencia del análisis y diseño de esta iteración se han realizado los siguientes refinamientos en los requisitos del proyecto.

Referencia	Requisito
R07.1	Si el usuario realiza el gesto para mover el sinóptico y sitúa su mano a la izquierda del punto de referencia el sinóptico se moverá a la izquierda
R07.2	Si el usuario realiza el gesto para mover el sinóptico y sitúa su mano a la derecha del punto de referencia el sinóptico se moverá a la derecha
R07.3	Si el usuario realiza el gesto para mover el sinóptico y sitúa su mano por encima del punto de referencia el sinóptico se moverá hacia abajo
R07.4	Si el usuario realiza el gesto para mover el sinóptico y sitúa su mano por debajo del punto de referencia el sinóptico se moverá hacia arriba

Tabla 7.1: Refinamiento de requisitos. Décima iteración

7.3. Implementación

En esta sección se describe la implementación del sistema de reconocimiento del usuario.

Una vez que se ha seleccionado un sinóptico se accede a una vista mayor de ese sinóptico pudiendo hacer zoom en él para ver mejor algún detalle o para alejarse y tener una vista general. Así mismo es posible mover el sinóptico para poder ver otros detalles cuando es muy grande y poder así navegar por la imagen.

Para poder moverse por el sinóptico se ha de estirar la mano entrando en el modo de movimiento del sinóptico. En ese momento si el usuario mueve su mano el sinóptico se moverá

según se especifica en los requisitos detallados en la sección 6.2.2. En la siguiente figura se muestra una imagen del usuario y una mira representando el punto de referencia que sirve para guiar al usuario en el movimiento que desea realizar en el sinóptico.

Figura 7.1: Punto de referencia para el movimiento de los sinópticos

Esto se consigue obteniendo la posición de la mano a través de los datos que ofrece el dispositivo Kinect y comparando este posición con el punto de referencia. El punto de referencia si el usuario utiliza la mano izquierda en lugar de la derecha se sitúa a la misma altura y profundidad en la parte izquierda del usuario.

En esta iteración también se ha creado una página que muestra una gráfica y una tabla con estadísticas de las últimas alarmas que se han producido en la central. La gráfica es una imagen que se toma de un servicio web aportado por la empresa. Los datos de la tabla son tomados de otro servicio web aportado también por la empresa.

En la siguiente figura se puede ver el resultado de la página creada.

Figura 7.2: Página de alarmas

7.5. Sumario

En este capítulo se ha descrito la décima y última iteración del desarrollo del cliente Kinect para IDbox.

Se detalla el proceso de desarrollo de esta iteración en la que se crea el sistema de movimiento de los sinópticos y la página que muestra la gráfica y la tabla con las estadísticas de las últimas alarmas. Se detallan los objetivos de la iteración, los casos de uso del sistema de reconocimiento, los requisitos que ha sido necesario añadir debido al análisis y el diseño de la iteración, la implementación del sistema y la verificación que se ha realizado para comprobar su correcto funcionamiento.

Capítulo 8

Conclusiones y trabajos futuros

El último capítulo de la memoria muestra las conclusiones que se han tomado del proyecto así como idea para posibles trabajos futuros en el proyecto.

Índice

8.1. Conclusiones

Esta sección detalla las conclusiones obtenidas de la realización del proyecto.

En este proyecto de fin de carrera se han investigado las posibilidades de la tecnología Kinect en el entorno nuclear buscando información sobre dicha tecnología y creando un pequeño proyecto como prueba de concepto para valorar si es posible su aplicación al entorno deseado.

Tras comprobar que la aplicación de este tecnología servía para los propósitos deseados en el entorno nuclear se ha desarrollado un cliente basado en dicha tecnología para el sistema de monitorización de plantas IDbox, sistema creado por la empresa CIC Consulting Informático.

Este cliente permite a los operarios de una central nuclear navegar obtener datos del sistema IDbox sin necesidad de contacto con ningún dispositivo para interactuar con el cliente.

La navegación por el cliente se efectúa a través de gestos corporales naturales para los operarios que no requieren prácticamente aprendizaje por su parte de manera que se sientan cómodos utilizando el cliente y no tengan problemas a la hora de manejarlo. Se navega por el sistema gracias a ocho gestos básicos que son: Saludar a la pantalla para tomar o dejar el control del cliente, realizar un movimiento de swype a derecha o a izquierda para avanzar o retroceder en

los elementos de un menú tanto horizontal como verticalmente, levantar la mano izquierda para seleccionar un elemento del menú, levantar la mano derecha para volver al menú anterior, estirar ambos brazos al frente para tomar el control del zoom cuando se está visualizando un sinóptico y estirar una mano al frente para tomar el control del movimiento del sinóptico en su visualización.

El cliente ofrece las siguientes opciones: Ver un cuadro de mando que muestra el estado general de la planta así como acceder a detalles sobre ese estado, acceder a un visor de sinópticos que nos permite seleccionar un sinóptico y verlo a mayor tamaño pudiendo hacer zoom sobre él o moverlo, ver una gráfica y una tabla con estadísticas sobre las últimas alarmas producidas en la planta y realizar consultas de los datos que brinda el sistema IDbox a través de un menú de categorías de dichos datos pudiendo ver el valor en tiempo real de cualquier dato así como una tabla o un gráfico con sus datos históricos.

Respecto a la etapa de investigación de este proyecto de fin de carrera he podido observar algunas de las herramientas que en un futuro podrán encontrarse en muchas empresas y que facilitarán el trabajo a los empleados gracias al uso de interfaces de usuario naturales que permiten trabajar de manera cómoda y rápida aumentando la efectividad del empleado así como su satisfacción personal en el desempeño de su trabajo.

Es muy importante en este aspecto fijar interacciones que sean realmente naturales para que la curva de aprendizaje de la aplicación sea prácticamente nula y los trabajadores se sientan realmente cómodos a la hora de utilizar la aplicación.

Durante la realización del cliente Kinect para IDbox he comprobado la importancia de un buen diseño del proyecto y de un buen análisis de los requisitos del mismo, siendo éstas fases fundamentales sobre las que se basa todo el proceso de desarrollo del proyecto y siendo muy costoso el cambio en alguna de las conclusiones tomadas en estas fases.

Gracias a la metodología iterativa e incremental utilizada en este proyecto se mitigan bastante los costes de un cambio en las fases fundamentales del proyecto pudiendo añadir nuevas funcionalidades en iteraciones adelantadas que no fueron planeadas al comienzo del proyecto. Aun así hay ciertos cambios en la estructura del proyecto que resultan muy costosos por lo que es necesario que el análisis principal del proyecto sea claro y conciso abarcando todos aquellos aspectos que se desean implementar.

8.2. Trabajos futuros

Esta sección se muestran los posibles trabajos futuros que tendrían cabida en el proyecto desarrollado si el proyecto tiene aceptación por parte de los usuarios finales.

Como trabajos de aplicación más inmediata se mejorará el proceso de dibujo del esqueleto sobre la imagen del usuario haciendo coincidir completamente las articulaciones representadas mediante puntos y líneas con las del propio usuario. Así mismo se incluirá en el cliente un apartado con vídeo explicativos de la interacción del usuario con la aplicación mostrando ejemplos

de los gestos que se utilizan para el control de ésta. También se investigará la implantación de un sistema por el cual si el usuario no está mirando a la pantalla no se detecten los gestos que haga para eliminar así posibles interacciones no deseadas con el sistema.

En un futuro más lejano se podrá definir un nuevo gesto más natural para la selección de las opciones de los menús como sería estirar el brazo simulando tocar la representación gráfica de dicha opción. También se investigarán las librerías de reconocimiento de gestos que durante las últimas fases del proyecto han sido publicadas por terceros, evaluando si son adecuadas para la sustitución del actual sistema de reconocimiento gestual liberando así de este trabajo a los futuros implicados en el proyecto.

Apéndice A

Contenidos del CD

En este apéndice se muestra el contenido del CD adjunto a esta memoria.

- **Memoria del proyecto:** Versión digital de esta memoria en formato *pdf*.

Bibliografía

[KNCT 2012] Página dedicada al dispositivo Kinect en Wikipedia. (12/09/2012)
http://es.wikipedia.org/wiki/Kinect

[IDb 2012] Página oficial de IDbox. (15/02/2012)
http://idbox.cic.es

[KfW 2012] Página oficial de Kinect para Windows (15/02/2012)
http://kinectforwindows.org

[DUMONT 2012] Tutorial de uso del seguimiento de esqueletos de Kinect por Renaud Dumont (09/09/2012)
http://www.renauddumont.be/en/2012/kinect-sdk-1-0-3-tracker-les-mouvements-avec-le-skeletonstream

[KBNM 2012] Introducción al *near mode* en el blog de Kinect for Windows. (09/09/2012)
http://blogs.msdn.com/b/kinectforwindows/archive/2012/01/20/near-mode-what-it-is-and-isn-t.aspx

[NATHAN 2006] Adam Nathan. *Windows Presentation Foundation Unleashed*. Primera edición, 2006.

[KX360 2012] Página oficial de Kinect para Xbox 360. (12/09/2012)
http://www.xbox.com/es-ES/Kinect

[KQSS 2012] Kinect For Windows Quickstart Series @ Channel 9 (10/03/2012)
http://channel9.msdn.com/Series/KinectQuickstart

[KQS1 2012] Tutorial de Kinect para Windows. Instalación del sensor (10/03/2012)
http://channel9.msdn.com/Series/KinectQuickstart/Installing-and-Using-the-Kinect-Sensor

[KQS2 2012] Tutorial de Kinect para Windows. Instalación del entorno de desarrollo (10/03/2012)
http://channel9.msdn.com/Series/KinectQuickstart/Setting-up-your-Development-Environment

[KQS3 2012] Tutorial de Kinect para Windows. Cámara RGB (10/03/2012)
http://channel9.msdn.com/Series/KinectQuickstart/Camera-Fundamentals

[KQS4 2012] Tutorial de Kinect para Windows. Sensor de profundidad (10/03/2012) http://channel9.msdn.com/Series/KinectQuickstart/Working-with-Depth-Data

[KQS5 2012] Tutorial de Kinect para Windows. Seguimiento del esqueleto (10/03/2012) http://channel9.msdn.com/Series/KinectQuickstart/Skeletal-Tracking-Fundamentals

[KQS6 2012] Tutorial de Kinect para Windows. Audio (10/03/2012) http://channel9.msdn.com/Series/KinectQuickstart/Audio-Fundamentals

[WMC 2012] Página de la Wikipedia sobre el modelo de desarrollo en cascada. (09/09/2012) http://es.wikipedia.org/wiki/Desarrollo_en_cascada

[BitSpe 2006] Kurt Bittner, Ian Spence. *Managing Iterative Software Development Projects*. Primera edición, 2006.

[RaGaAnMi 2010] Nick Randolph, David Gardner, Chris Anderson, Michael Minutillo. *Professional Visual Studio 2010*. Primera edición, 2010.

[RIBAS 2010] Joan Ribas Lequerica. *Web Services*. Primera edición, 2010.

[CHROME 2012] Página oficial del navegador Google Chrome. (12/09/2012) http://www.google.com/chrome?hl=es

[N++ 2011] Página oficial del bloc de notas Notepad++. (12/09/2012) http://notepad-plus-plus.org/

[MuTri 2007] Pedro Muñoz Rodriguez, Bienvenido Trillo Sáez. *Iniciación a Gimp*. Primera edición, 2007.

[SiStBo 2010] Carl Siechert, Craig Stinson, Ed Boot. *El libro de Windows 7*. Primera edición, 2010.

[ROEBUCK 2011] Kevin Roebuck. *.Net Framework: High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*. Primera edición, 2011.

[MOORE 2010] Andrew Moore. *Create Amazing Custom User Interfaces with Wpf, C#, and Xaml in .Net 3.0*. Primera edición, 2010.

[MILES 2012] Rob Miles. *Using Kinect for Windows with XNA*. Primera edición, 2012.

[BRUNO 2012] Bolg de Bruno Capuano. Desarrollador. (09/09/2012)
http://elbruno.com http://geeks.ms/blogs/elbruno

[STCKOF 2012] Stack Overflow. Foro de dudas para programadores.
http://stackoverflow.com

[CH9NUIBB 2012] Charla sobre Interfaces de Usuario Naturales con Bill Buxton. (05/04/2012)
http://channel9.msdn.com/Blogs/LarryLarsen/CES-2010-NUI-with-Bill-Buxton

[REE 2011] Informe del Sistema Eléctrico Español. (07/09/2012)
http://www.ree.es/sistema_electrico/informeSEE.asp

[OKP 2012] The Open Kinect Project. Concurso lanzado por Adafruit Industries (09/09/2012)
http://www.adafruit.com/blog/2010/11/04/the-open-kinect-project-the-ok-prize-get-1000-bounty-for-kinect-for-xbox-360-open-source-drivers/

[KM 2012] Vídeo de demostración de Kinect Mouse. (09/09/2012)
http://www.youtube.com/watch?v=GkLMUooozZs

[KP 2012] Vídeo de demostración del Kinect Paint (09/09/2912)
http://www.youtube.com/watch?v=2VTAz-o1W1E

[KFD 2012] Recopilación de las diferencias entre Kinect para Xbox 360 y Kinect para Windows (09/09/2012)
http://blog.kinectfordevelopers.com/2012/03/01/diferencias-entre-kinect-xbox-360-y-kinect-for-windows/

[CDP 2012] Codeplex. Sitio de alojamiento para proyectos de código abierto. (09/09/2012)
http://codeplex.com

[KMC 2012] Proyecto Kinect Mouse Cursor en Codeplex. (09/09/2012)
http://kinectmouse.codeplex.com