De [Fernando](https://www.blogger.com/profile/01876040281567664622) - [abril 10, 2012](https://cortesfernando.blogspot.com/2012/04/sdk-kinect-for-dummies-1.html)

Todos conocemos Kinect, unos lo habréis probado en casa de algún amigo, otros solo lo habéis visto en la tele. Sea como sea Kinect nos puede servir como herramienta de desarrollo y de investigación o simplemente como entretenimiento. Al poco de aparecer Kinect hubo un SDK no oficial para poder emplear Kinect en nuestros PCs. Pasado un tiempo Microsoft lanzó el SDK de Kinect para Windows. Pero en los detalles de hardware o como ha ido evolucionando Kinect no es en lo que me fijaré en el blog. Intentaré orientarme más a la programación.

[](https://1.bp.blogspot.com/-ZUdGQuE6BRE/T4SciaC_adI/AAAAAAAAAGU/8ettMWF1NrQ/s1600/kinect.jpg)

Ahora viene mi opinión. Pues bien me he puesto con el SDK de Kinect y me gusta. He estado haciendo cosillas con el reconocimiento de voz aunque de momento no tengo material para subir ninguna entrada. Por otra parte como primer paso con el SDK de Kinect veremos como pintar lo que muestra la cámara. Es decir, tan solo la cámara sin Skeleton ni nada por el estilo.

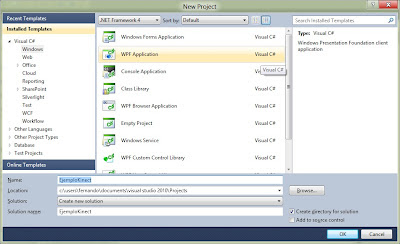
Necesitamos:

- Visual Studio: versión Express o si somos estudiantes desde [www.dreamspark.com](http://www.dreamspark.com/)

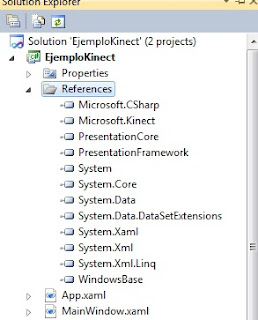
- SDK de Kinect: <http://www.microsoft.com/en-us/kinectforwindows/develop/overview.aspx>

- Sensor Kinect.

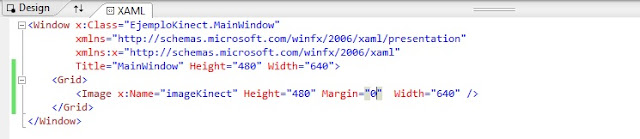
Una vez tenemos todo podemos comenzar a familiarizarnos con el SDK de Kinect. Abrimos Visual Studio y creamos un nuevo proyecto: 'WPF application'. Le damos un nombre a nuestro proyecto, en mi caso lo he llamado ejemploKinect.

[](https://1.bp.blogspot.com/-Ugsh3_-xKb4/T4ScnU6YAoI/AAAAAAAAAGk/AhVU3iCYM_s/s1600/nuevo.jpg)

Una vez agregado el proyecto debemos agregar la referencia de Kinect a nuestro proyecto. Vamos al Solution Explorer y en nuestro proyecto en 'Reference' hacemos clic derecho y seleccionamos 'Add Reference..' y en la pestaña de .NET buscamos la **referencia Microsoft.Kinect**.

[](https://4.bp.blogspot.com/-pe2eugU5fRE/T4ScoVJ_eUI/AAAAAAAAAGs/wIsDjj1g7mw/s1600/ReferenciaKinect.jpg)

Una vez agregada la referencia debemos ir al *"MainWindow.xaml"* y agregar un control del tipo imagen. Le cambiaremos el nombre al que queramos y modificaremos su ancho y su alto. Debido a que la imagen de Kinect va a 640x480 recomiendo poner 480 de alto y 640 de ancho. Debido a que no me funciona correctamente el script para subir código mostraré una imagen.

[](https://2.bp.blogspot.com/-E9j7oQUB028/T4ScpJJQElI/AAAAAAAAAG0/znV-V1N2hh4/s1600/xaml.jpg)

A continuación vamos al .cs asociado a MainWindow. Esto es "MainWindow.xaml.cs". El código esta comentado por tanto no será necesario explicaciones extra. Tan solo apuntar que se debe añadir "**using Microsoft.Kinect;**"

///

/// Interaction logic for MainWindow.xaml

///

public partial class MainWindow : Window

{

KinectSensor miKinect;

public MainWindow()

{

InitializeComponent();

// Agregamos el evento de carga de la ventana

Loaded += MainWindowLoaded;

}

void MainWindowLoaded(object sender, RoutedEventArgs e)

{

// Comprobamos que tenemos un sensor conectado

if (KinectSensor.KinectSensors.Count > 0)

{

//Evento ejecutado al cerrar

Closing += MainClosing;

// Escogemos el primer sensor kinect que tengamos conectado. Puede haber más de un kinect conectado

miKinect = KinectSensor.KinectSensors[0];

// Habilitamos la cámara elegiendo el formato de imagen.

miKinect.ColorStream.Enable(ColorImageFormat.RgbResolution640x480Fps30);

// Arrancamos Kinect.

miKinect.Start();

// Nos suscribimos al método

miKinect.AllFramesReady += KinectAllFramesReady;

}

}

///

/// Obtenemos los frames y los pintamos en la imagen.

///

///

///

void KinectAllFramesReady(object sender, AllFramesReadyEventArgs e)

{

//Obtenemos el frame de imagen de la camara

using (var colorFrame = e.OpenColorImageFrame())

{

// Si este es null no continuamos

if (colorFrame == null) return;

// Creamos un array de bytes del tamaño de los pixel del frame.

byte[] pixels = new byte[colorFrame.PixelDataLength];

//Copiamos los pixel del frame a nuestro array de bytes.

colorFrame.CopyPixelDataTo(pixels);

// Colocamos los pixel del frame en la imagen del xml

int stride = colorFrame.Width \* 4;

imageKinect.Source = BitmapSource.Create(colorFrame.Width, colorFrame.Height, 96, 96, PixelFormats.Bgr32, null, pixels, stride);

//imageKinect es el objeto imagen colocado en el xaml

}

}

void MainClosing(object sender, EventArgs e)

{

// Al cerrar paramos Kinect

if (miKinect != null)

{

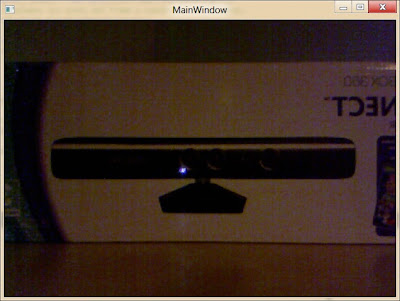
miKinect.Stop();

}

}

}

Y ahora veamos como queda:

[](https://3.bp.blogspot.com/-Eh1VsxDx6JI/T4Sck3YhTOI/AAAAAAAAAGc/ElN9GWYhdSE/s1600/finEjemplo.jpg)

**Las imágenes se ven como un espejo** debido a que Kinect esta preparado para que nosotros nos veamos en él.

Intentaré subir el proyecto proximamente.

[C#](https://cortesfernando.blogspot.com/search/label/C%23) [Kinect](https://cortesfernando.blogspot.com/search/label/Kinect)

**Comentarios**

1. https://resources.blogblog.com/img/blank.gif

**Anónimo**[*1 de febrero de 2013, 6:46*](https://cortesfernando.blogspot.com/2012/04/sdk-kinect-for-dummies-1.html?showComment=1359697603195#c2168483833641899923)

Gracvias amigo, esto me interesa y es bueno que hayas puesto un ejemplo como este de claro

[**RESPONDER**](javascript:;)

1. https://www.blogger.com/img/blogger_logo_round_35.png

[**Tony**](https://www.blogger.com/profile/05424141607032937704)[*2 de marzo de 2013, 0:11*](https://cortesfernando.blogspot.com/2012/04/sdk-kinect-for-dummies-1.html?showComment=1362179518409#c3979681986503918506)

muy buen aporte yo tambien acabo de iniciarme a la programacion de mismo

[**RESPONDER**](javascript:;)

1. https://2.bp.blogspot.com/-p9PQwHMS1lQ/T4xHtE_fy4I/AAAAAAAAAHI/5gQ2PtTedDQ/s35/Imagen001.jpg

[**GianMarkin**](https://www.blogger.com/profile/15323762398713241750)[*14 de abril de 2013, 2:59*](https://cortesfernando.blogspot.com/2012/04/sdk-kinect-for-dummies-1.html?showComment=1365901159402#c996194404835952174)

Muy interesante, un buen inicio para comenzar a programar en Kinect.. gracias por el tuto :)

[**RESPONDER**](javascript:;)

1. https://www.blogger.com/img/blogger_logo_round_35.png

[**Unknown**](https://www.blogger.com/profile/03005866685428080532)[*23 de septiembre de 2013, 6:55*](https://cortesfernando.blogspot.com/2012/04/sdk-kinect-for-dummies-1.html?showComment=1379912157861#c7509875376194344514)

namespace WpfApplication1  
{  
///  
/// Lógica de interacción para MainWindow.xaml  
///  
public partial class MainWindow : Window  
{  
Microsoft.Kinect. KinectSensor miKinect;  
public MainWindow()  
{  
InitializeComponent();  
// Agregamos el evento de carga de la ventana  
Loaded += MainWindowLoaded;  
}  
  
void MainWindowLoaded(object sender, RoutedEventArgs e)  
{  
// Comprobamos que tenemos un sensor conectado  
if (Microsoft.Kinect.KinectSensor.KinectSensors.Count > 0)  
{  
//Evento ejecutado al cerrar  
Closing += MainClosing;  
// Escogemos el primer sensor kinect que tengamos conectado. Puede haber más de un kinect conectado  
miKinect =Microsoft.Kinect.KinectSensor.KinectSensors[0];  
// Habilitamos la cámara elegiendo el formato de imagen.  
miKinect.ColorStream.Enable(Microsoft.Kinect.ColorImageFormat.RgbResolution640x480Fps30);  
// Arrancamos Kinect.  
miKinect.Start();  
// Nos suscribimos al método  
miKinect.AllFramesReady += KinectAllFramesReady;  
}  
  
}  
  
void KinectAllFramesReady(object sender, Microsoft.Kinect.AllFramesReadyEventArgs e)  
{  
//Obtenemos el frame de imagen de la camara  
using (var colorFrame = e.OpenColorImageFrame())  
{  
// Si este es null no continuamos  
if (colorFrame == null) return;  
// Creamos un array de bytes del tamaño de los pixel del frame.  
byte[] pixels = new byte[colorFrame.PixelDataLength];  
//Copiamos los pixel del frame a nuestro array de bytes.  
colorFrame.CopyPixelDataTo(pixels);  
// Colocamos los pixel del frame en la imagen del xml  
int stride = colorFrame.Width \* 4;  
imageKinect.Source = BitmapSource.Create(colorFrame.Width, colorFrame.Height, 96, 96, PixelFormats.Bgr32, null, pixels, stride);  
//imageKinect es el objeto imagen colocado en el xaml  
}  
}  
  
void MainClosing(object sender, EventArgs e)  
{  
// Al cerrar paramos Kinect  
if (miKinect != null)  
{  
miKinect.Stop();  
}  
}  
}  
}  
  
  
codigo en vb2012