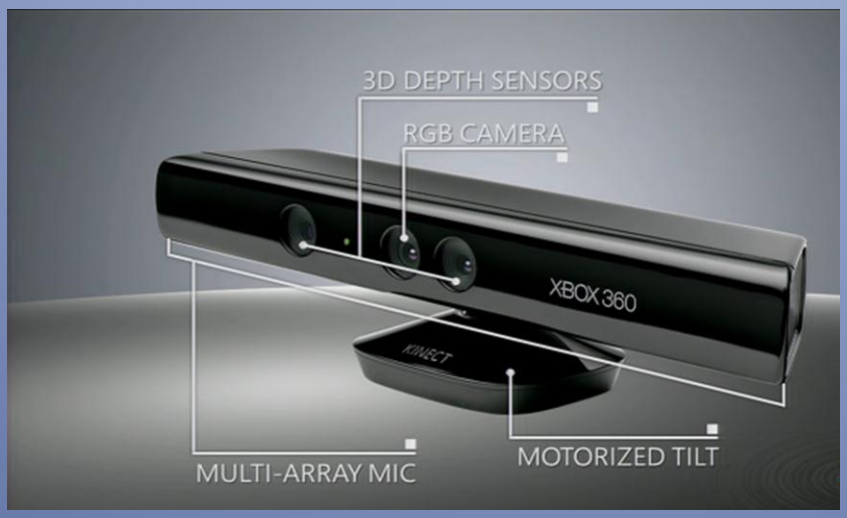


Nuevos Paradigmas de la Interacción

Practica 1 – Movimiento 30

Desplazar el pie izquierdo en el eje -X una distancia determinada en un parámetro de entrada.

Realizado por:

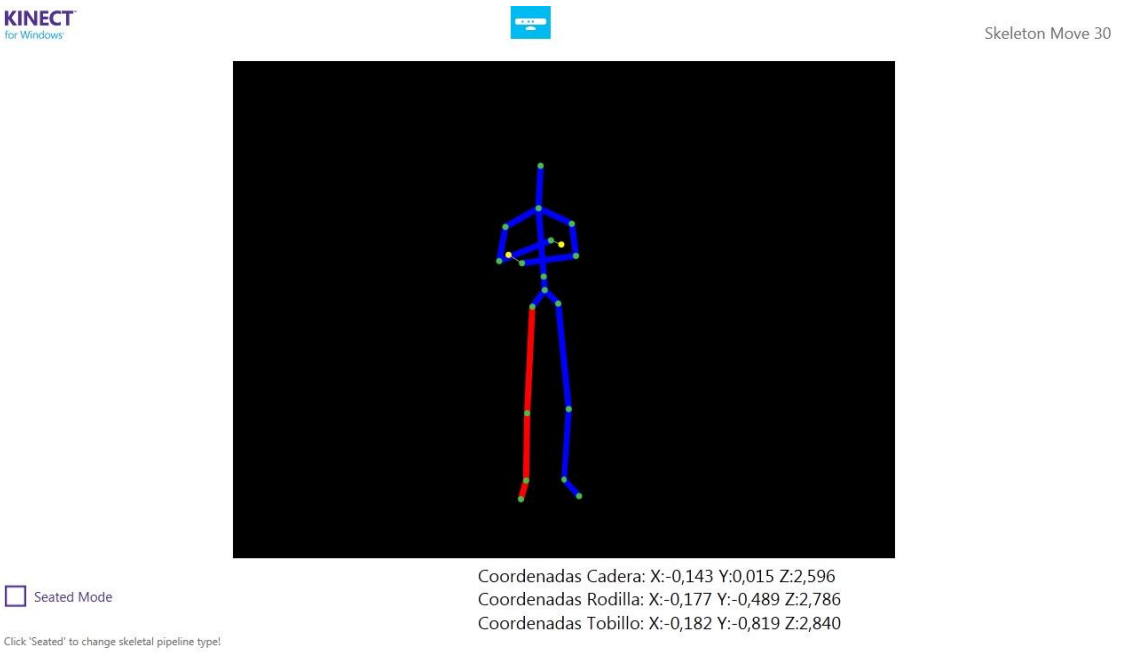
José Delgado Dolset

Descripción del programa:

La clase se compone de un booleano, que se usa para comprobar si se ha colocado correctamente la pierna inicialmente, para iniciar el movimiento, y dos flotantes, que indican la distancia necesaria a recorrer en el eje x y la tolerancia en las medidas, respectivamente. También se han implementado un constructor (que inicializa estos tres atributos), una función para comprobar que estamos en la posición inicial y otra que se comprueba que se hace correctamente el movimiento.

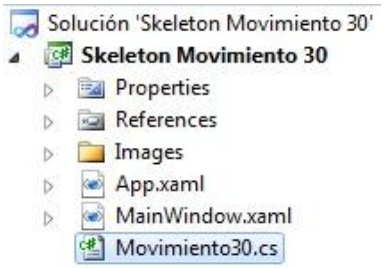
En la clase principal, se han declarado dos pinceles para pintar la pierna izquierda en dos colores distintos, uno para cuando está bien colocada y otro para cuando está mal colocada. También se ha modificado la función de dibujo del esqueleto, para que no se dibuje la pierna izquierda, y se ha añadido otra que la pinta en los colores correspondientes.

Por último, se ha modificado la pantalla de interfaz para que se muestren las coordenadas de las articulaciones de la pierna izquierda.



Instrucciones de Uso de la Clase

1) Lo primero es colocar el archivo “Movimiento30.cs” en la carpeta correspondiente del proyecto de Visual Studio. Hay que tener en cuenta que usa el namespace “namespace Microsoft.Samples.Kinect.SkeletonBasics” así que si no lo está ya, habrá que añadir las directivas correspondientes.



2) Hecho esto, hay que declarar un objeto **global** de la clase “Movimiento30”.

```
/// <summary>
/// The object we'll use to controlling the move
/// </summary>
private Movimiento30 checkMove;
```

3) Este objeto hay que inicializarlo. Se recomienda llamar al constructor en la ventana principal, donde se inicializan todos los parámetros iniciales. Es necesario pasarle como parámetros los dos flotantes que representan la distancia en el eje –X que debe recorrer el esqueleto y la tolerancia, respectivamente.

```
/// <summary>
/// Initializes a new instance of the MainWindow class.
/// </summary>
public MainWindow()
{
    InitializeComponent();
    checkMove = new Movimiento30(0.75f, 0.05f);
}
```

4) Una vez hecho esto, hay que añadir al manejador del SkeletonStream la llamada a la función **`bool movimientoCorrecto(Skeleton skel),`** que devolverá positivo si se ha realizado correctamente el movimiento. Se puede utilizar esta información como se desee. En nuestro caso, lo único que se hace es pintar de distinto color la pierna izquierda, según se haya realizado correctamente o no el movimiento.

```
if (skel.TrackingState == SkeletonTrackingState.Tracked)
{
    if (this.checkMove.movimientoCorrecto(skel))
    {
        this.DrawBone2(skel, dc, JointType.HipLeft, JointType.KneeLeft, this.correctPositionPen);
        this.DrawBone2(skel, dc, JointType.KneeLeft, JointType.AnkleLeft, this.correctPositionPen);
        this.DrawBone2(skel, dc, JointType.AnkleLeft, JointType.FootLeft, this.correctPositionPen);
    }
    else
    {
        this.DrawBone2(skel, dc, JointType.HipLeft, JointType.KneeLeft, this.badPositionPen);
        this.DrawBone2(skel, dc, JointType.KneeLeft, JointType.AnkleLeft, this.badPositionPen);
        this.DrawBone2(skel, dc, JointType.AnkleLeft, JointType.FootLeft, this.badPositionPen);
    }
    this.DrawBonesAndJoints(skel, dc);
    leftAnkle = skel.Joints[JointType.AnkleLeft];
    leftKnee = skel.Joints[JointType.KneeLeft];
    leftHip = skel.Joints[JointType.HipLeft];

    coordenadas = string.Format("Coordenadas Cadera: X:{0:0.000} Y:{1:0.000} Z:{2:0.000}", leftHip.Position.X, leftHip.Position.Y, leftHip.Position.Z);
    coordenadas += string.Format("\nCoordenadas Rodilla: X:{0:0.000} Y:{1:0.000} Z:{2:0.000}", leftKnee.Position.X, leftKnee.Position.Y, leftKnee.Position.Z);
    coordenadas += string.Format("\nCoordenadas Tobillo: X:{0:0.000} Y:{1:0.000} Z:{2:0.000}", leftAnkle.Position.X, leftAnkle.Position.Y, leftAnkle.Position.Z);
}
```

Bugs y Errores Detectados:

En las pruebas realizadas, se ha comprobado que hay que añadir una tolerancia significativa para que funcione correctamente, lo cual nos indica que la precisión del dispositivo no es todo lo buena que debería ser. Es posible que se pueda mejorar el algoritmo empleado para el cálculo de la posición, pero no se han hecho pruebas por falta de tiempo.