

MOVIMIENTO 31

Mover la mano derecha en el eje -X una distancia determinada en un parámetro de entrada.

Introducción

- ▶ El movimiento a realizar consiste en, a partir de la postura de inicio 4, llevar el brazo derecho hacia atrás.
- ▶ Es decir, manteniendo los brazos rectos en cruz llevaremos la mano hacia atrás sobre el eje Z todo lo que nos sea posible.

Solución realizada I: Variables

- Variables necesarias para saber en que gesto/posición nos encontramos y pintar los huesos.

```
public enum Posture
{
    None,
    Inicio,
    Correcto,
    Transcurso,
};
const int PostureDetectionNumber = 10;
int accumulator = 0;
Posture postureError = Posture.None;
Posture postureStart = Posture.Inicio;

// Puntos de union y Pens con los que vamos a pintar los huesos del cuerpo segun el movimiento
Joint wristR, elbowR, shoulderR;
private readonly Pen penCorrecto = new Pen(Brushes.Green, 6);
private readonly Pen penTranscurso = new Pen(Brushes.Yellow, 6);
private readonly Pen penInicio = new Pen(Brushes.Blue, 6);
private readonly Pen penError = new Pen(Brushes.Red, 6);

// Booleanos para controlar la posicion y pintar los huesos de distinto color
private bool reposo = false;
private bool proceso = false;
private bool correcto = false;
```

Solución realizada II: Inicio

- Cogemos los puntos de unión y llamamos a la comprobación de gestos solamente si el punto del hombro se encuentra en tracked.

```
foreach (Skeleton bones in skeletons)
{
    // Guardamos los puntos de union que nos interesan para el movimiento
    if (bones.TrackingState == SkeletonTrackingState.Tracked)
    {
        wristR = bones.Joints[JointType.WristRight]; //MUÑECA
        elbowR = bones.Joints[JointType.ElbowRight]; //CODO
        shoulderR = bones.Joints[JointType.ShoulderRight]; //HOMBRO
    }
}

// Llamada a las comprobaciones de la posicion del brazo,
// para que acceda el punto del hombro debe estar en tracking
// sino produce errores.
if (shoulderR.TrackingState == JointTrackingState.Tracked)
    comprobarGestos(wristR, elbowR, shoulderR);
```

Solución realizada III: Métodos

- ▶ Comprueba si el brazo se encuentra en la posición inicial (en cruz).

```
public bool PosInicio(Joint wristR, Joint elbowR, Joint shoulderR)
```

- ▶ Comprueba si se ha empezado a realizar el ejercicio.

```
public bool TransMovimiento(Joint wristR, Joint elbowR, Joint shoulderR)
```

- ▶ Comprueba si se ha alcanzado la posición final.

```
public bool MovFinalizado(Joint wristR, Joint elbowR, Joint shoulderR)
```

- ▶ Comprueba cualquier caso de error y reinicia el ejercicio.

```
public bool CasoError(Joint wristR, Joint elbowR, Joint shoulderR)
```

Solución realizada IV:

- ▶ Método que llama sucesivamente a los métodos anteriores para controlar el movimiento
- ▶ 1º Comprueba la postura inicial

```
public void comprobarGestos(Joint wristR, Joint elbowR, Joint shoulderR)
{
    if (PosInicio(wristR, elbowR, shoulderR))
    {
        if (PostureDetector(Posture.Inicio))
        {
            solucionP.Content = "Postura de inicio correcta";
            reposo = true;
        }
    }
}
```

Solución realizada V:

- ▶ 2º Si no se encuentra en la postura inicial comprueba si ya paso por ella.

```
else
{
    // La primera postura que debe reconocer sera la de reposo sino
    // no dara comienzo el ejercicio.
    if (reposo)
    {
        if (MovFinalizado(wristR, elbowR, shoulderR))
        {
            if (PostureDetector(Posture.Correcto))
            {
                correcto = true;
                proceso = false;
                solucionP.Content = "Movimiento finalizado correctamente";
            }
        }
        else if (TransMovimiento(wristR, elbowR, shoulderR))
        {
            if (PostureDetector(Posture.Transcurso))
            {
                proceso = true;
                solucionP.Content = "Mueva la mano hacia atras";
            }
        }
    }
}
```

```
else if (PosInicio(wristR, elbowR, shoulderR))
{
    if (PostureDetector(Posture.Inicio))
    {
        solucionP.Content = "Postura de inicio, comienze";
    }
    else if (CasoError(wristR, elbowR, shoulderR))
    {
        if (PostureDetector(Posture.None))
        {
            correcto = false;
            proceso = false;
            reposo = false;
            solucionP.Content = "Establezca la posicion inicial";
        }
    }
}
```

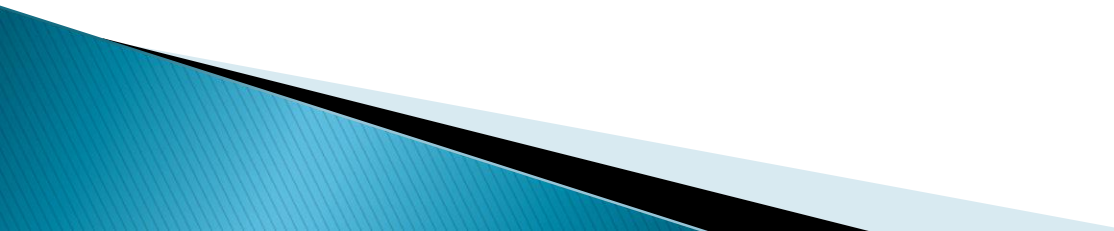
Solución realizada VI:

- Por ultimo para cambiar el color de los huesos según la posición.

```
// Pintamos el hueso Hombro - Codo segun la posicion en la que se encuentra
if (jointType0 == JointType.ShoulderRight && jointType1 == JointType.ElbowRight)
{
    drawPen = selectColor();
}
// Pintamos el hueso Codo - Muñeca segun la posicion en la que se encuentra
else if (jointType0 == JointType.ElbowRight && jointType1 == JointType.WristRight)
{
    drawPen = selectColor();
}
// Pintamos el hueso Muñeca - Mano segun la posicion en la que se encuentra
else if (jointType0 == JointType.WristRight && jointType1 == JointType.HandRight)
{
    drawPen = selectColor();
}
// Pintamos el resto de huesos con el color por defecto
else
    drawPen = this.trackedBonePen;
```

```
public Pen selectColor()
{
    if (reposo)
    {
        if (proceso)
            return penTranscurso;
        else if (correcto)
            return penCorrecto;
        else
            return penInicio;
    }
    else
        return penError;
}
```


Problemas encontrados

- ▶ El eje de coordenadas y como aumentan o disminuyen los valores de los puntos según cada movimiento.
 - ▶ Ejecutar la solución solo si el esqueleto esta en tracked, ya que inicia con todos los valores a 0.
 - ▶ Descubrir la clase Joint y su funcionamiento.
- 

Bibliografía

- ▶ [1] <http://blogs.msdn.com/b/esmsdn/archive/2011/08/09/reto-sdk-de-kinect-detectar-poses-con-skeletal-tracking.aspx>
- ▶ [2] <http://msdn.microsoft.com/en-us/library/>