

Práctica 2. Nuevo Paradigmas de Interacción



Hugo Mario Lupión Fernández
Antonio Doncel Campos

Ejercicio fitness que realiza una sentadilla mientras el usuario cierra los brazos.

30/11/2014

Descripción de la práctica

La realización de esta práctica lleva implementado el ejercicio 3, que es la realización de una sentadilla, y el ejercicio 31 invertido, que es cerrar los brazos estirados partiendo de la posición de tener los brazos rectos formando un ángulo de 180 grados con los hombros.

Hemos optado por la realización en conjunto de los dos movimientos, impidiendo que el mismo usuario pueda realizar un ejercicio y después otro.

Para realizarlo debe el usuario hacer a la misma vez, una sentadilla y los brazos cerrarlos al mismo tiempo, partiendo de la posición de reposo, terminando el ejercicio en un tiempo y haciéndolo a una velocidad media, para que el programa no lo detecte por mal.

Utilizando las herramientas que nos da Kinect, hemos utilizado el detector de puntos, además del pintado de los puntos del esqueleto, para dar la información al usuario si está realizando bien o mal el movimiento. Se va pintando el esqueleto primeramente en rojo, dando una advertencia de que no se está en ninguna posición correcta, cambiando al color azul, si se está en la posición inicial de la que se parte, pasando a un color amarillo mientras se está realizando bien el ejercicio, hasta llegar un color verde si se ha realizado correctamente.

Se tendrá un contador con los movimientos correctos que se han realizado, además de un botón con el error que el usuario quiera meter (rango entre 5% de error y 15% máximo), habiéndosele puesto un rango medio para que pueda realizar bien el ejercicio. También se tiene un botón de reiniciar todo, por si el usuario quiere volver a empezar el ejercicio, además de ayudas visuales como que movimientos tiene que hacer para realizar una serie completa.

Módulos de la práctica.

Inicialmente se cuenta con dos módulos principales que son los que realizan todas las llamadas.

El modulo uno (función), comprueba que el movimiento parte de la posición inicial correcta establecida y se va realizando bien el ejercicio, es decir hacer la sentadilla con el cierre de brazos a la vez.

Este módulo comprueba que el usuario no haga movimiento extraños que no estén establecidos.

Cuando haga toda las comprobaciones y el usuario ha realizado el movimiento completo, saltara al módulo dos (otra función) que lo que hace es el ejercicio a la inversa, volviendo a empezar cuando llegue al final del ejercicio, habiendo contado como el ejercicio por bueno, cuando haya terminado cada módulo correctamente.

Código principal de la práctica

```
//POSICION INICIAL

/*****
*****/

/*****
*****/

// Comprueba si se encuentra en la posición inicial
// Mano derecha en cruz y piernas rectas.
public bool BrazosRectos(Joint wristR, Joint elbowR, Joint shoulderR,
Joint wristL, Joint elbowL, Joint shoulderL, float error)
{
    if (elbowR.Position.Y < shoulderR.Position.Y + 0.06 + (0.03 * error)
    && elbowR.Position.Y > shoulderR.Position.Y - 0.06 + (0.03 * error) &&
        wristR.Position.Y < shoulderR.Position.Y + 0.1 + (0.05 * error)
    && wristR.Position.Y > shoulderR.Position.Y - 0.1 + (0.05 * error) &&
        elbowR.Position.Z < shoulderR.Position.Z + 0.06 + (0.03 * error)
    && elbowR.Position.Z > shoulderR.Position.Z - 0.15 + (0.1 * error) &&
        wristR.Position.Z < shoulderR.Position.Z + 0.06 + (0.03 * error)
    && wristR.Position.Z > shoulderR.Position.Z - 0.2 + (0.15 * error))
    {
        if (elbowL.Position.Y < shoulderL.Position.Y + 0.06 + (0.03 *
error) && elbowL.Position.Y > shoulderL.Position.Y - 0.06 + (0.03 * error) &&
            wristL.Position.Y < shoulderL.Position.Y + 0.1 + (0.05 *
error) && wristL.Position.Y > shoulderL.Position.Y - 0.1 + (0.05 * error) &&
            elbowL.Position.Z < shoulderL.Position.Z + 0.06 + (0.03 *
error) && elbowL.Position.Z > shoulderL.Position.Z - 0.15 + (0.1 * error) &&
            wristL.Position.Z < shoulderL.Position.Z + 0.06 + (0.03 *
error) && wristL.Position.Z > shoulderL.Position.Z - 0.2 + (0.15 * error))
            {
                return true;
            }
        }
    }
    return false;
}

public bool PiernasRectas(Joint hipC, Joint kneeR, Joint kneeL, Joint
ankleR, Joint ankleL)
{
    if ((kneeL.Position.Z - ankleL.Position.Z) - (hipC.Position.Z -
kneeL.Position.Z) > 0.005f - (0.01f * error) &&
        (kneeR.Position.Z - ankleR.Position.Z) - (hipC.Position.Z -
kneeR.Position.Z) > 0.005f - (0.01f * error))
    {
        ankleIR = ankleR;
        ankleIL = ankleL;
        return true;
    }
    return false;
}

public bool PosInicio(Joint wristR, Joint elbowR, Joint shoulderR, Joint
wristL, Joint elbowL, Joint shoulderL, Joint hipC, Joint kneeR, Joint kneeL,
Joint ankleR, Joint ankleL, float error)
{
    if (BrazosRectos(wristR, elbowR, shoulderR, wristL, elbowL,
shoulderL, error) && PiernasRectas(hipC, kneeR, kneeL, ankleR, ankleL))
        return true;

    return false;
}
```

```

    }

    /*****
    *****/

    /*****
    *****/

    // FASE1

    /*****
    *****/

    /*****
    *****/

    // Comprueba si está avanzando en la hacia la posición final de la Fase1
    del ejercicio.
    // Mano hacia delante hasta quedar en paralelo.
    public bool TransMovimiento1(Joint wristR, Joint elbowR, Joint shoulderR,
    Joint wristL, Joint elbowL, Joint shoulderL, float error)
    {
        if (elbowR.Position.Z < shoulderR.Position.Z - 0.07 + (0.07 * error)
        && wristR.Position.Z < shoulderR.Position.Z - 0.07 + (0.07 * error))
        {
            if (elbowL.Position.Z < shoulderL.Position.Z - 0.07 + (0.07 *
            error) && wristL.Position.Z < shoulderL.Position.Z - 0.07 + (0.07 * error))
            {
                return true;
            }
        }
        return false;
    }

    // Comprueba si ha llegado a la posición final de la Fase1.
    // Brazos en paralelo delante del cuerpo y en horizontal al suelo.
    public bool Fase1Final(Joint wristR, Joint elbowR, Joint shoulderR, Joint
    wristL, Joint elbowL, Joint shoulderL, float error)
    {
        if (elbowR.Position.Z < shoulderR.Position.Z - 0.20 + (0.20 * error)
        && wristR.Position.Z < shoulderR.Position.Z - 0.20 + (0.20 * error))
        {
            if (elbowL.Position.Z < shoulderL.Position.Z - 0.20 + (0.20 *
            error) && wristL.Position.Z < shoulderL.Position.Z - 0.20 + (0.20 * error))
            {
                return true;
            }
        }
        return false;
    }

    /*****
    *****/

    /*****
    *****/

    // FASE2

    /*****
    *****/

```

```

/*****
*****/
// Comprueba si esta avanzando en la hacia la posición final de la Fase2
del ejercicio.
// Bajar poco a poco flexionando las rodillas y con la espalda recta.
public bool TransMovimiento2(Joint hipC, Joint kneeR, Joint kneeL, Joint
ankleR, Joint ankleL, float error)
{
    if ((hipC.Position.Y - kneeR.Position.Y) < 0.70 + (0.70 * error) &&
(hipC.Position.Y - kneeL.Position.Y) < 0.70 + (0.70 * error) &&
        (hipC.Position.Y - kneeR.Position.Y) > 0.53 + (0.53 * error) &&
(hipC.Position.Y - kneeL.Position.Y) > 0.53 + (0.53 * error))
    {
        return true;
    } //Esto o <
    return false;
}

// Comprueba si ha llegado a la posición final de la Fase2.
// Brazos en paralelo delante del cuerpo y en horizontal al suelo en
sentadilla.
public bool Fase2Final(Joint hipC, Joint kneeR, Joint kneeL, Joint
ankleR, Joint ankleL, float error)
{
    if ((hipC.Position.Y - kneeR.Position.Y) < 0.52 + (0.52 * error) &&
(hipC.Position.Y - kneeL.Position.Y) < 0.52 + (0.52 * error))
    {
        return true;
    }
    return false;
}

/*****
*****/

/*****
*****/

// ERROR

/*****
*****/

/*****
*****/

// Comprobación de los casos de error.
// Reiniciar el ejercicio si se cumple cualquiera de ellos.
public bool CasoError(Joint wristR, Joint elbowR, Joint shoulderR, Joint
wristL, Joint elbowL, Joint shoulderL, Joint hipC, Joint kneeR, Joint kneeL,
Joint ankleR, Joint ankleL, Joint ankleIR, Joint ankleIL, float error)
{
    if (elbowR.Position.Y > shoulderR.Position.Y + 0.06 + (0.06 * error)
|| elbowR.Position.Y < shoulderR.Position.Y - 0.06 + (0.06 * error) ||
        wristR.Position.Y > shoulderR.Position.Y + 0.08 + (0.08 * error)
|| wristR.Position.Y < shoulderR.Position.Y - 0.08 + (0.08 * error) ||
        elbowL.Position.Y > shoulderL.Position.Y + 0.06 + (0.06 * error)
|| elbowL.Position.Y < shoulderL.Position.Y - 0.06 + (0.06 * error) ||
        wristL.Position.Y > shoulderL.Position.Y + 0.08 + (0.08 * error)
|| wristL.Position.Y < shoulderL.Position.Y - 0.08 + (0.08 * error) ||

```

```

        ankleR.Position.Y > ankleIR.Position.Y + 0.06 + (0.06 * error) ||
ankleL.Position.Y > ankleIL.Position.Y + 0.06 + (0.06 * error))
    {
        return true;
    }

    return false;
}

/*****
*****/

/*****
*****/

```

Estas son las funciones principales con las que se realiza todo el ejercicio.

En primer lugar partimos de la posición inicial que comprueba si está el usuario en la posición definida, cuando la cumple pasa a transcurso de movimiento mientras no haga ningún movimiento raro.

Por ultimo si se ha cumplido lo anterior entraría en fase final que comprueba que se ha llegado a la posición establecida final.

Como concatenamos dos movimientos volvemos a hacer lo mismo de antes pero a la inversa, es decir, la posición final del primero es la posición inicial del segundo y de la misma manera con la posición final del segundo y la posición de inicio del primero.

Problemas encontrados

Principalmente el mayor problema aparte de los ya encontrados con cada movimiento, ha sido la concatenación de los dos ejercicios, para que los realice correctamente, sin que se salga de los rangos establecidos, ya que para realizar el ejercicio correctamente, el usuario debe realizar una sentadilla y mientras la va realizando también debe ir cerrando los brazos al mismo tiempo, ya que si realiza un ejercicio más rápido que otro, se dará por invalido el ejercicio y contara como mal ese movimiento.