INTERFAZ DE USUARIO PARA ROBOT REHABILITADOR DE MANO

primer anÁlisis

PABLO horno pérez

2017

## Objetivo

Diseñar y programar un interfaz de usuario, donde apoyar el control y funcionalidad del robot de rehabilitación de mano, renderizar un modelo de una mano en tiempo real con los datos obtenidos por los actuadores lineales y representar cada uno de los dedos en la posición correspondiente implementando las ecuaciones reunidas en los cálculos del proyecto.

Se requiere de un protocolo de comunicación entre la máquina que soportara el interfaz y el controlador que gestiona el movimiento de los dedos.

Debemos diferenciar 2 partes en este proyecto:

1. **Interfaz de usuario**, entendemos como IU todo el conjunto de recursos que se le ofrece al usuario para manipular los datos del programa y ejecución de ejercicios de la mano de rehabilitación.
2. **Ventana de gráficos,** este es el encargado de representar la mano o el exoesqueleto, embebido en el interfaz de usuario o en una ventana aparte.

## Base inicial

Partimos de un microcontrolador arduino mega, el cual dispone de 16 entradas analógicas y 54 E/S digitales, de los cuales, quince disponen de salida PWM.

Para la estimulación de los dedos contamos con cinco actuadores lineales, y los tipos de movimientos se seleccionan con la combinación de dos interruptores.

## POSIBLES SOLUCIONES

Las soluciones propuestas siguen el siguiente esquema de funcionamiento, donde un proceso es el encargado de la comunicación entre el Controlador Arduino y la Ventana de renderizacion.

### C# y processing

Emplear la librería de Windows Forms en C# para generar la interfaz de usuario y encargarse del control y comunicación con el arduino, podemos agregar botones, campos de texto, directorios de trabajo donde guardar información recibida y lanzar una aplicación encargada de la renderizacion del modelo 3D.

El lenguaje C# cuenta con una clase para el trabajo con entradas y salidas del sistema operativo Windows, podemos utilizar el puerto serial para la comunicación con arduino de forma muy sencilla, así como gestionar los eventos del interfaz gráfico y crear paquetes de comunicación para enviarlos tanto al arduino como a la ventana de render.

Processing es un entorno de programación en C++ de código abierto el cual nos permite la renderizacion de objetos 3D y manipulación de los mismos a través del puerto Serial.

### C# e Inventor API

Aprovechándonos de Windows Forms y el esquema anterior podemos emplear la API de inventor para generar una ventana de renderizacion y editar nuestro modelo 3D.

**Ventajas:** Fácil creación del modelo 3D de la mano, modificaciones de parámetro, asi como movimientos de cámara y demás propiedades del modelo.

**Inconvenientes:** Requiere de una sesión iniciada en Autodesk Inventor para poder visualizar piezas, lo que no es muy rentable.

### Processing

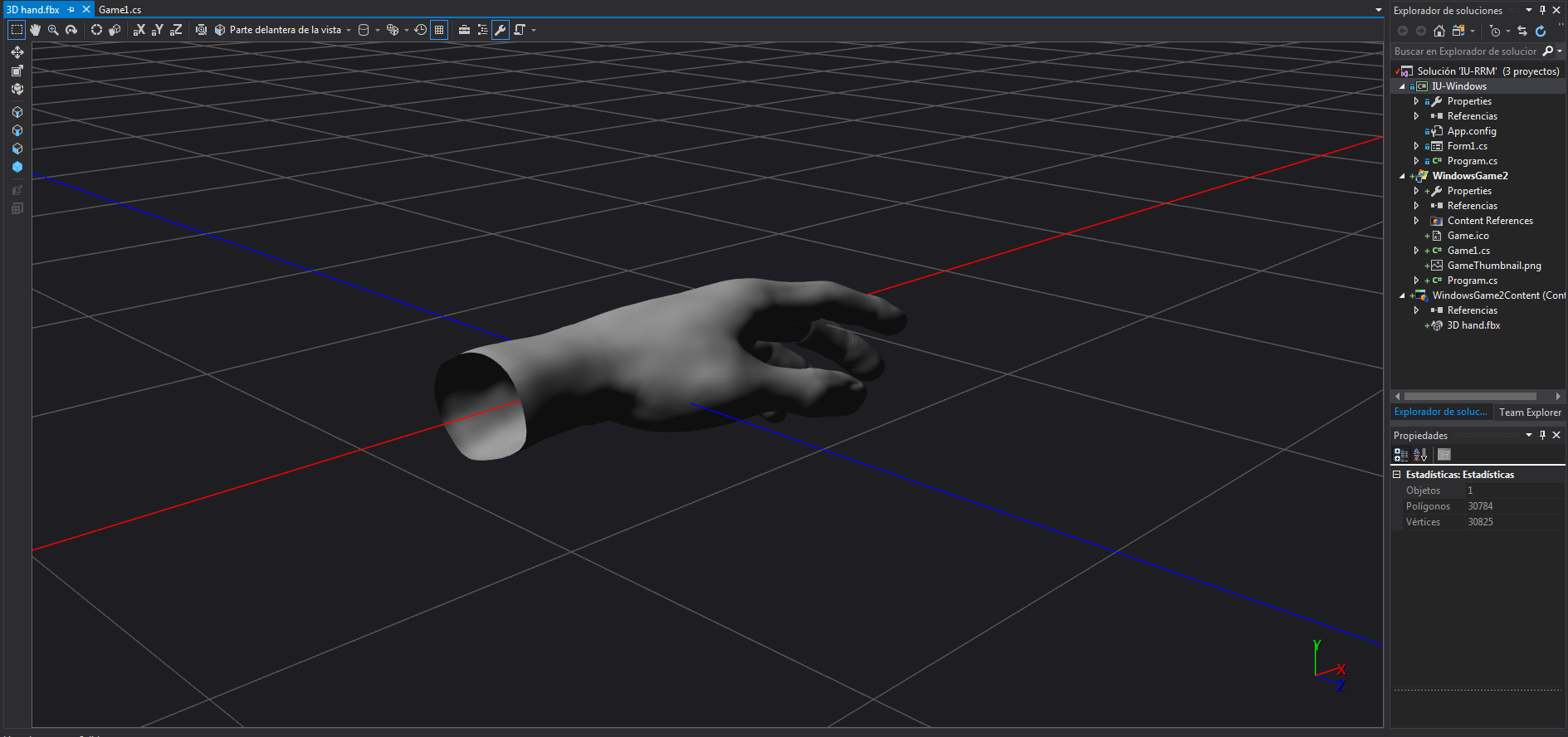
Ya que processing nos permite la comunicación con el puerto serial podemos programar, en C++, el interfaz con botones renderizados en pantalla, asignarles eventos y controlar los movimientos de la mano mediante comunicación Serial, directamente con el controlador Arduino.

**Inconvenientes** de esta opción es no disponer de un soporte sólido para la creación del interfaz de usuario, debiendo crear los botones y herramientas para interactuar con el usuario, así como sus eventos y disparadores.

**Ventajas** un solo lenguaje de programación, C++, una única comunicación serial.

### C# y XNA Game Studio

C# dispone de una librería para crear escenas 3D, modificadores de luz y demás instrucciones para el uso de objetos 3D, empleando el motor gráfico de XNA. La ventana de renderizacion y el interfaz de usuario se programan integrados en la misma ventana y proyecto.



**Ventajas** no es necesario una comunicación serial entre el modelo gráfico y el interfaz de usuario, podemos reutilizar un gráfico de una mano de [SuperDasil](https://www.blendswap.com/blends/view/81285), editar los huesos de la matriz de puntos que conforman la mano con blender.

**Inconvenientes** necesidad de crear una escena control de cámara y luces, así como los huesos para articular la mano, aun así, gran parte del modelo ya está creado