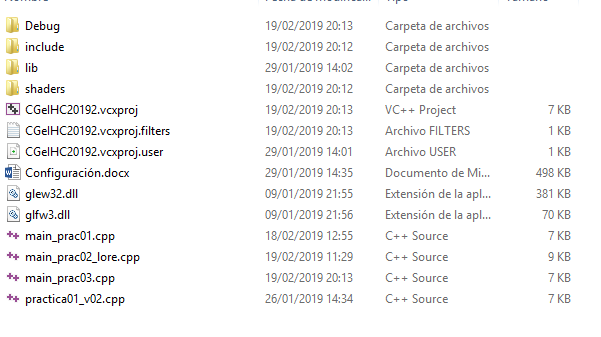
Práctica 5 – Transformaciones Geométricas

Para esta práctica el archivo comprimido tiene:

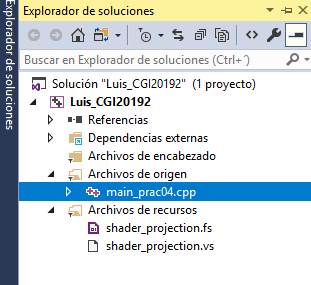
* Main\_prac05.cpp
* Carpeta “shaders”

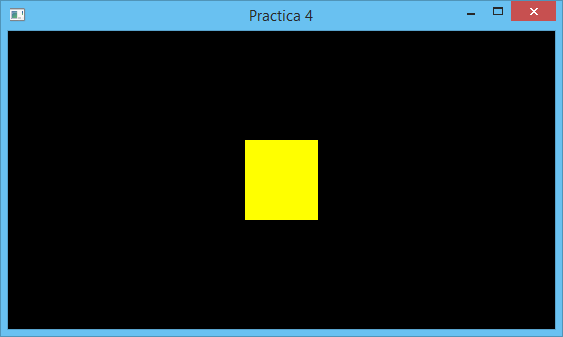
\*\*\*La carpeta “shaders” contiene el shader que permite asignar un color de forma individual a cada uno de los cubos. Se supone que en la práctica 4 se debió haber construido. La práctica 4 hace uso de esa modificación de ese shader.

Se debe descomprimir en la carpeta del proyecto (si pide sobre escribir archivos indicar que Sí) y debe quedar así la estructura:



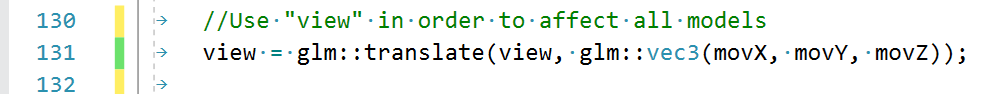
En Visual Studio, en el Explorador de Soluciones, se debe agregar el main\_prac05.cpp y quitar el de prácticas pasadas,



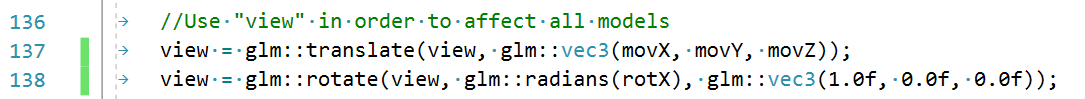
El código no debería de dar problemas y lo que se debe apreciar al ejecutar el programa es:

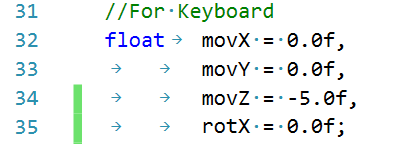
En la práctica 3 ya le había enseñado la operación de Traslación. Esta operación se sigue realizando con las teclas W, A, S, D, Av Pag y Re Pág.

Ahora les enseñaré la transformación geométrica de Rotación, así que al código de dibujo:

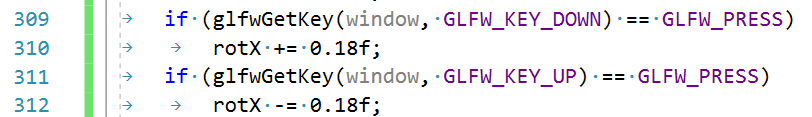


Se le adiciona una rotación, en mi caso puse una rotación con una variable sobre el eje X:

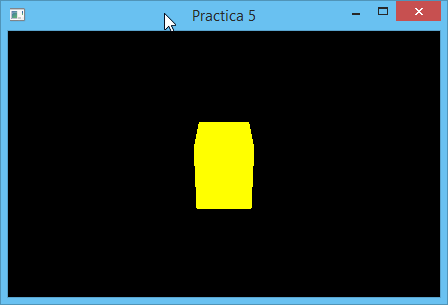


Recordar que se debe declarar una variable global:

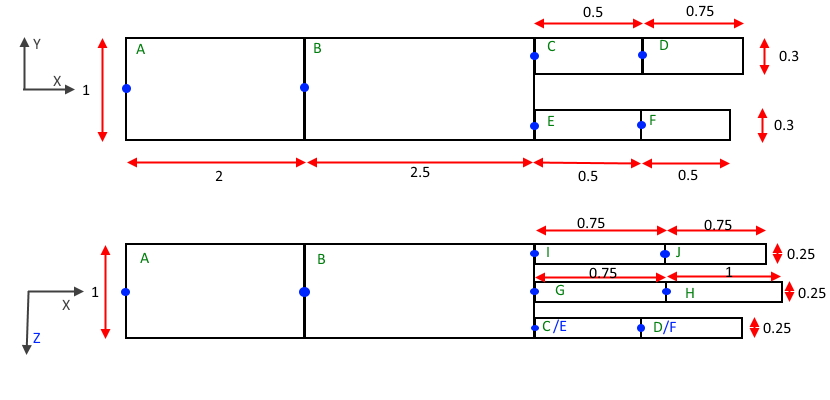
Y se le asigna teclas para modificar la variable, yo utilizaré las teclas de flechas arriba y abajo, dentro de la función *my\_input()*:

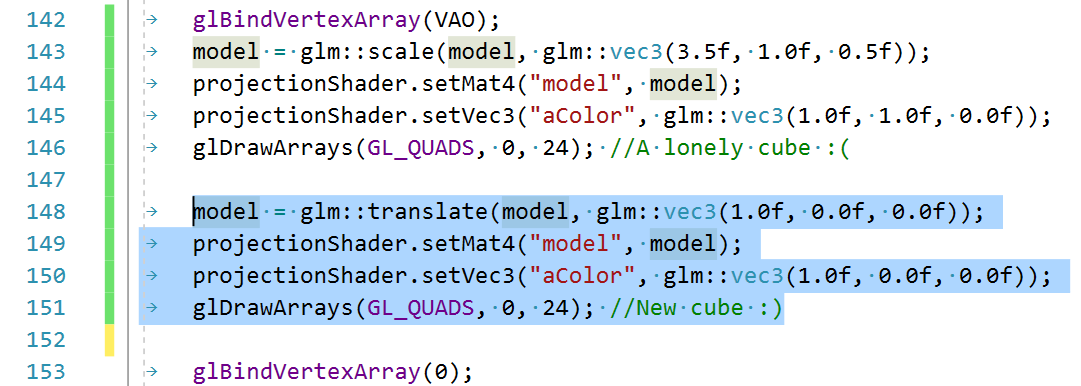


El resultado es:



El objetivo de la práctica es crear el brazo robot:





Algunas observaciones:

Por la complejidad del elemento se tienen dos vistas, primero está la vista frontal (ejes X, Y) y la vista superior (ejes X, Z).

El brazo robot tendrá asociadas algunas “animaciones”, las cuales consisten en las articulaciones del elemento, en este caso representadas por los puntos de color azul, simulando ser los movimientos del hombro, codo, y de los dedos.

Debido a estas mismas articulaciones, se debe definir la jerarquía de la construcción, es decir, la forma en que se transmite el movimiento. Poniendo la analogía del brazo humano podemos ver que al mover la articulación del hombro, ésta articulación afecta a todos los elementos del brazo, por lo cual se le considera de mayor jerarquía, en cambio, al mover una articulación que corresponda a los dedos, los elementos que son afectados son menos, por lo cual está más abajo en la jerarquía. Esto se representa por el siguiente diagrama jerárquico.

C

D

F

E

G

H

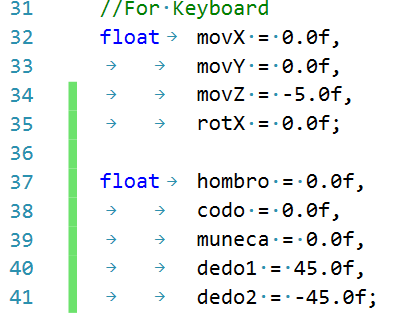
I

J

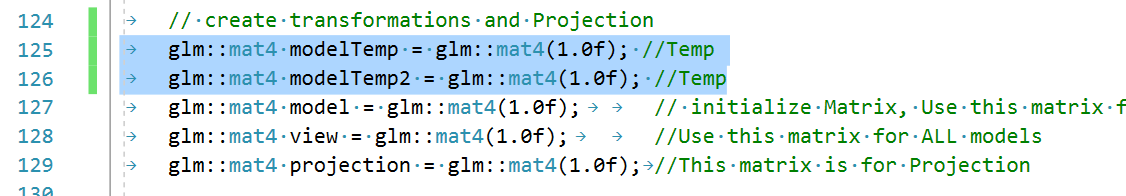
B

A

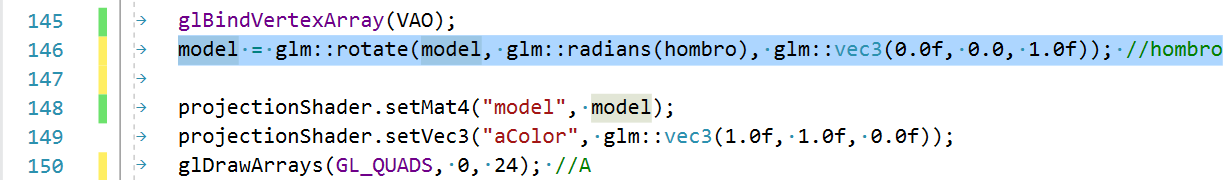
Recomiendo comenzar declarando las variables que servirán para las rotaciones (hombro, codo, etc.) como globales:



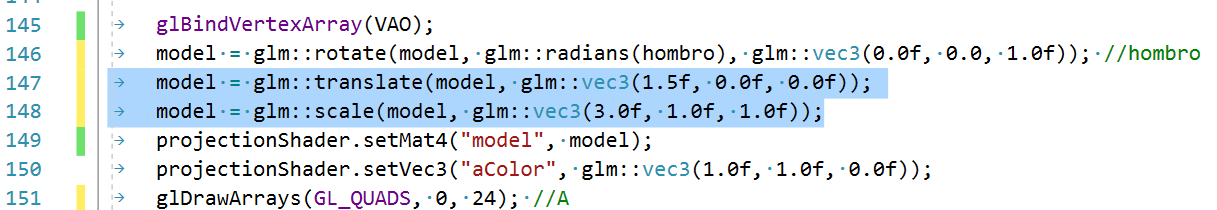
Para trabajar con Jerarquía, YO recomiendo utilizar matrices temporales que ayudarán a determinar el lugar a partir del cual vamos realizar las operaciones, según yo, para el brazo necesitamos dos matrices temporales, las creo en la función *display()*:



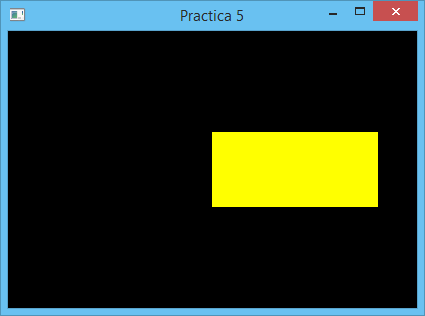
Comienzo la construcción de la figura con la rotación que simulará ser el **Hombro**, para ello creo una rotación sobre el eje Z:



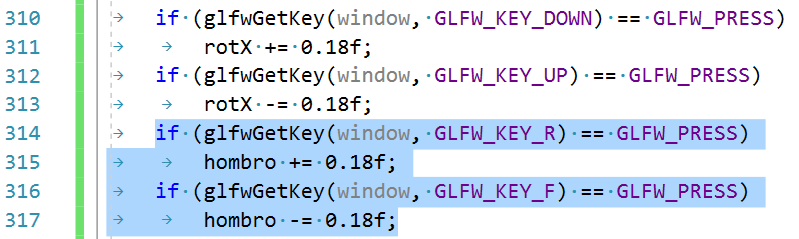
Ya con la rotación que simula ser la articulación, procedo a construir mi objeto A, para ello debo llegar a su centro geométrico y aplicar la escala:



Con eso obtengo la figura:



Falta hacer que la articulación se mueva con ayuda de una tecla, para ello se debe ir ala función de *my\_Input():*

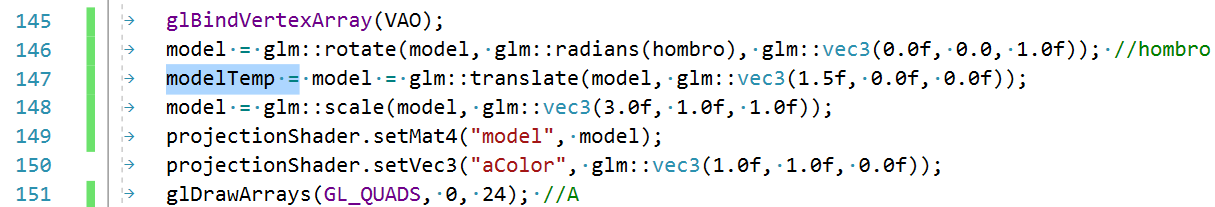


Ahora viene la parte, según yo, importante y difícil de explicar, que es el uso de la matriz temporal.

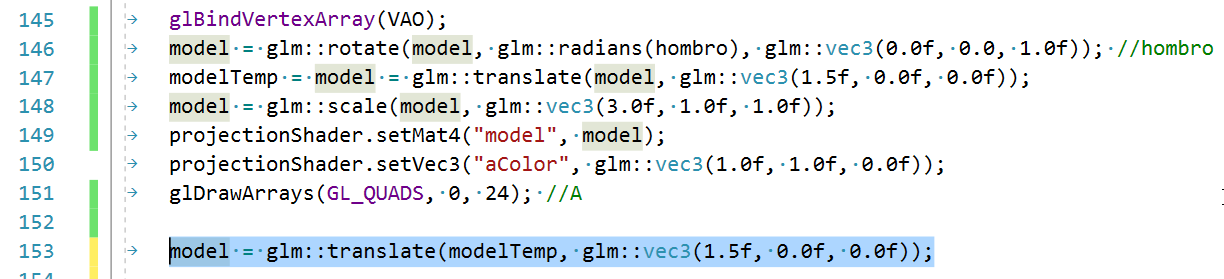
Se supone que en la sesión anterior vieron que las operaciones se acumulan y que también se puede “resetear”, utilizando la matriz identidad, sin embargo, ahora no nos interesa:

1. Conservar la escala de la Figura A, ni
2. Regresar al origen.

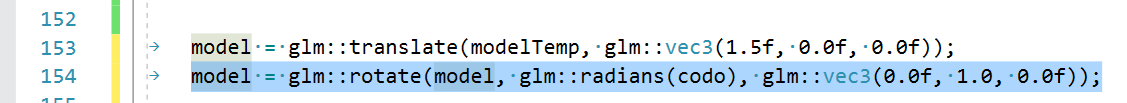
Lo que necesitamos es **Conservar la rotación del hombro**, por lo cual la solución es utilizar una matriz temporal en un valor donde se tenga ya aplicada la rotación del hombro. En mi caso se me hace sencillo el conservar hasta donde llegamos al centro de la figura A, ya que es antes de la escala pero después de la rotación, así que ahí utilizaré la primera matriz temporal:



Así que la construcción del siguiente elemento lo hago desde esa posición así que pongo en el código:

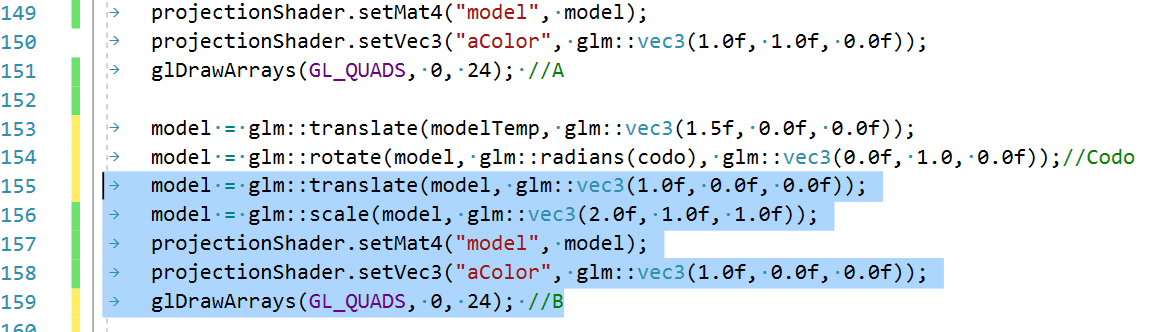


Esa traslación, la hago desde la primera matriz temporal y me ayuda a ubicarme entre la figura A y B, para ahí colocar la articulación del **Codo:**

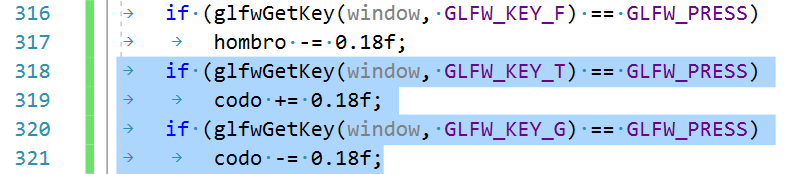


Dicha articulación quiero que gire sobre el eje Y.

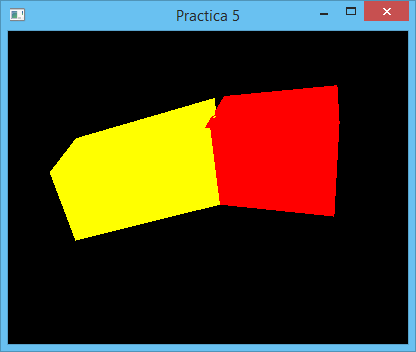
Ya con la articulación del codo, procedo a moverme al centro de la figura B, colocar su escala y dibujar:



La variable **codo** se debe modificar también desde teclado:

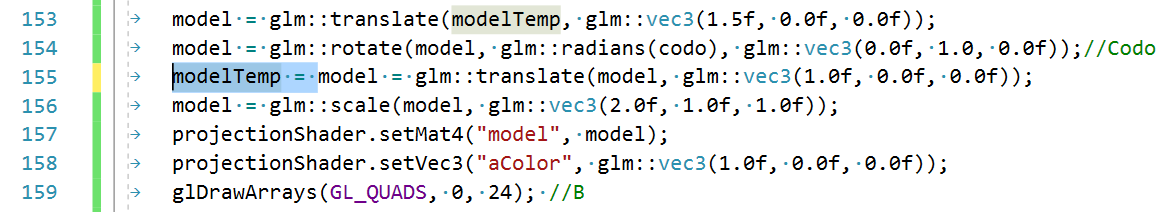


Hasta aquí se obtiene:

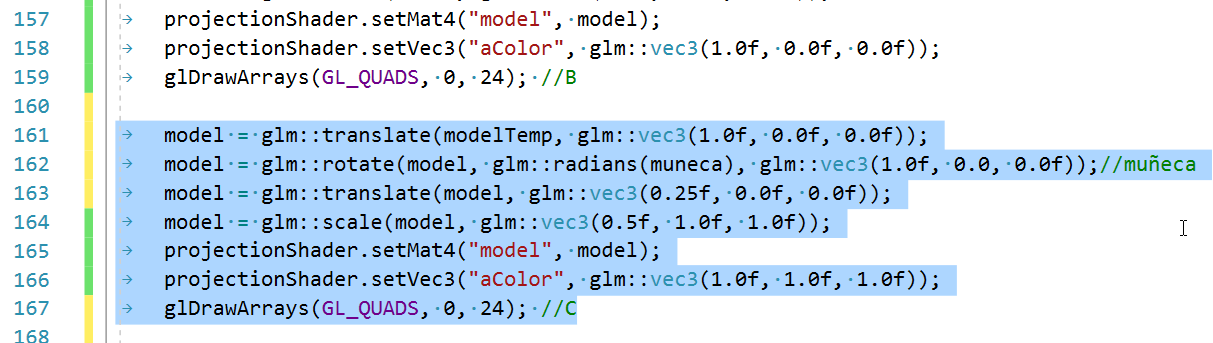


**Recomendación:** invitar a que los alumnos limiten el movimiento del *codo* y *hombro*, hombre hasta 90 grados hacia arriba y 110 grados hacia abajo. El codo 90 grados hacia adentro, y 5 grados hacia afuera.

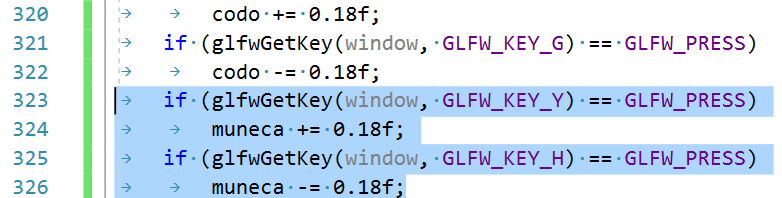
Se vuelve a utilizar la primera matriz temporal, para determinar el lugar desde el cual se desea iniciar las siguientes operaciones, yo recomiendo que sea antes de la escala de la figura B, y después de la rotación del codo, así que propongo:



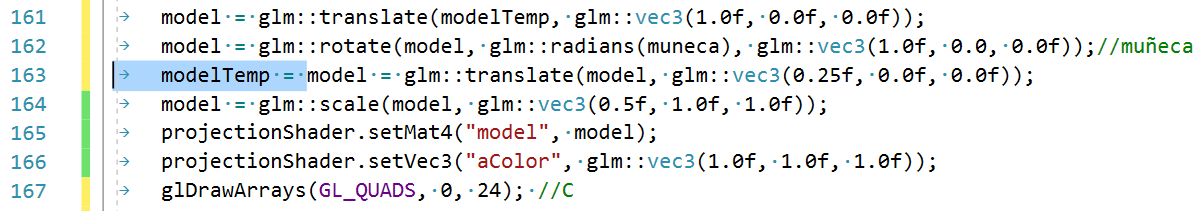
Realizo algo parecido para el elemento C, es decir, me muevo al punto donde quiero la rotación, coloco la rotación (**muneca**), me muevo al centro de C, aplico la escala y dibujo:



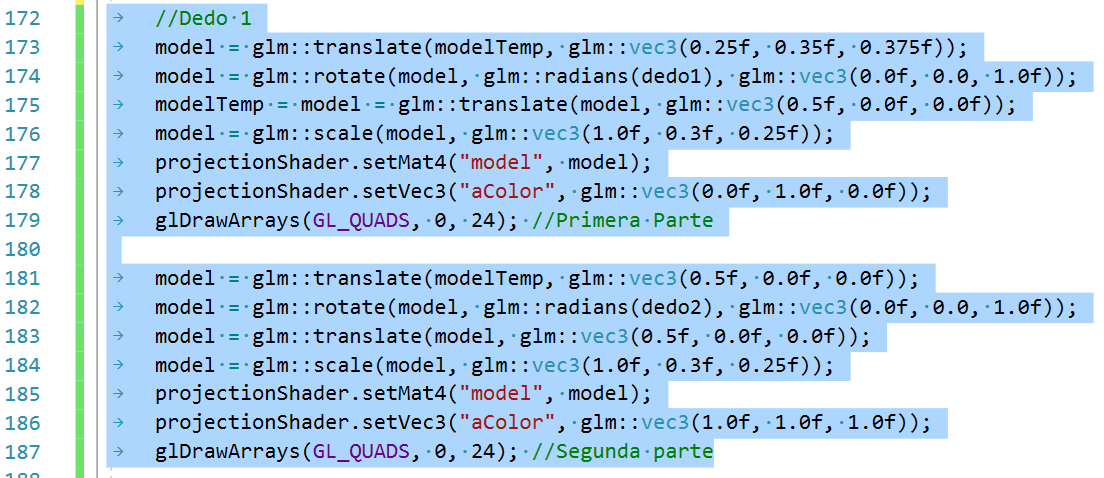
A la variable **muñeca** debo modificarla desde teclado:



Vuelvo a utilizar mi primera matriz temporal:

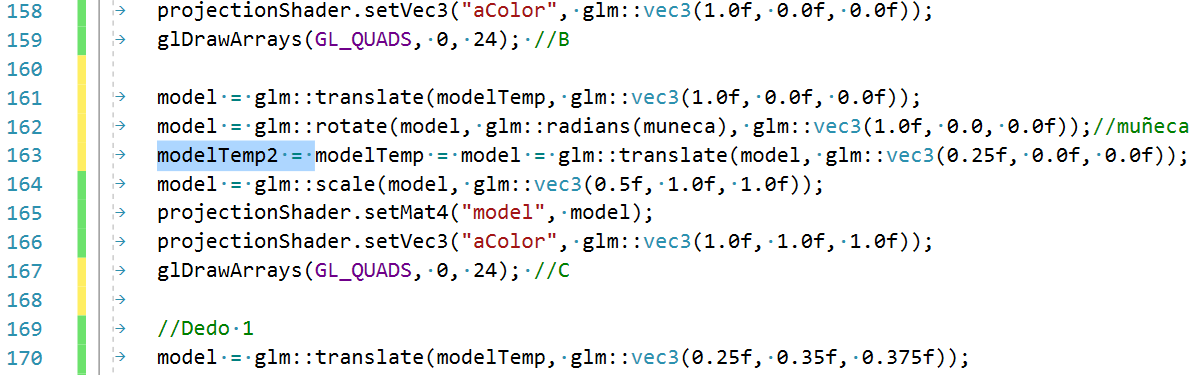


Y construyo dedos:

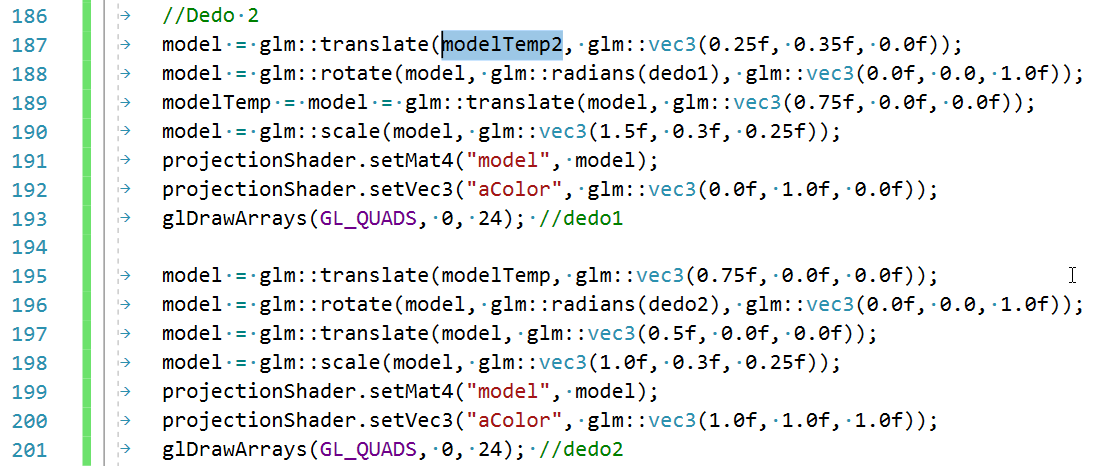


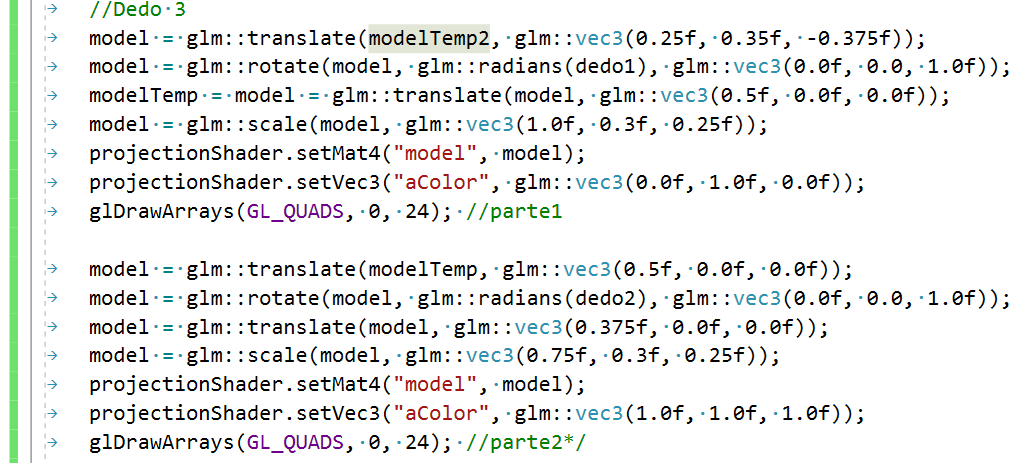
**Importante 2:** Aquí viene el uso de la segunda matriz temporal. ¿Por qué no utilizar la primera matriz temporal? La primera matriz temporal, si se revisa el código, se utilizó para construir la segunda parte del dedo a partir de la primera parte. Si para construir el siguiente dedo se utiliza la primera matriz temporal lo que estaríamos haciendo es que el segundo dedo dependerá de la primer parte del primer dedo. *Espero que esto no suene confuso.*

Como los dedos son independientes entre sí, o al menos eso se intentaría, se debe construir el siguiente dedo, sin tomar en cuenta nada del primer dedo, así que yo recomiendo hacer:



Retornar hasta el centro de la muñeca, y ahí colocar ambas matrices temporales, y recuperar la matriz temporal dos, en la construcción de los demás dedos, es decir algo como:





Me queda:

