

# FAQ

Sistemas Gráficos

Grado en Ingeniería Informática

Curso 2019/2020

## 1. Pregunta

Me habéis planteado una duda relacionada con el ejercicio 4 de prácticas. En concreto está relacionada con el hecho de cómo conseguir las rotaciones de las figuras que se ven en el vídeo.

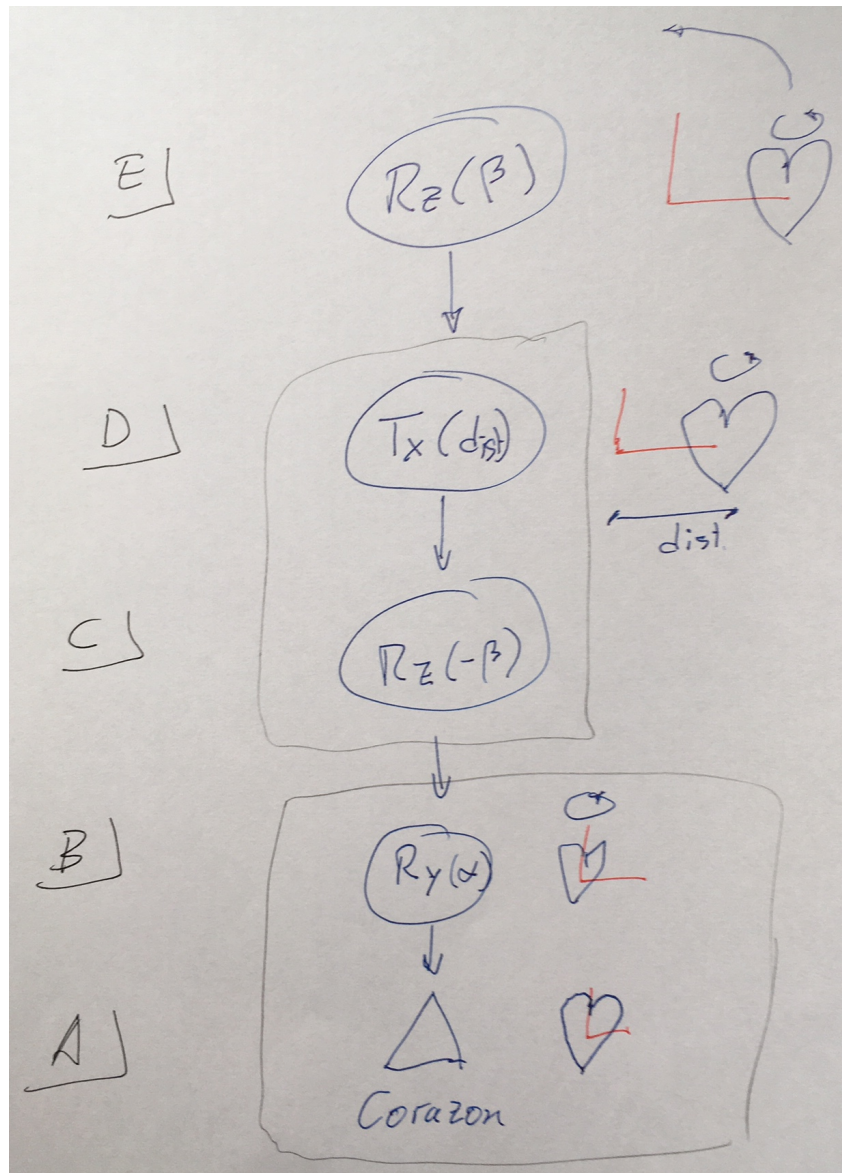
El corazón, por ejemplo, va girando sobre su eje vertical al mismo tiempo que, a una determinada distancia, va haciendo un giro circular con respecto a un centro, al mismo tiempo que el corazón mantiene su verticalidad.

## 2. Respuesta

Para conseguir que el corazón tenga los giros que se muestran en el vídeo hay que montar una jerarquía que sale con más nodos de los que inicialmente podíamos pensar. Pero intervienen varias rotaciones y una traslación que deben ir bien ordenadas para que el resultado sea el deseado.

Seguir la explicación viendo la foto del grafo que se acompaña. Las letras A, B, ... E. Son los nombres que les he puesto a los nodos para hacer la explicación.

Sigamos el grafo desde abajo hacia arriba.



Nodo A.

Es el Mesh con el que hemos construido el corazón. **IMPORTANTE:** El corazón debe estar más o menos centrado con el sistema de referencia, si no, el corazón no girará sobre sí mismo. Habrá salido más o menos centrado dependiendo de los puntos que hayáis usado para definir el contorno antes de hacer la extrusión.

Si no está centrado no hay mayor problema, ya que solo habría que trasladarlo para que esté centrado.

Por ejemplo:

## FAQ

---

```
var contornoCorazon = new THREE.Shape();
// aquí irán las instrucciones para crear el contorno

// a partir del contorno se construye la geometría
var geometriaCorazon = new THREE.ExtrudeGeometry (contornoCorazon, opciones);

// ahora es cuando trasladamos la geometría para que el corazón quede centrado,
// claro que lo mejor es dibujar ya el contorno centrado
geometriaCorazon.translate ( lo que corresponda para que esté centrado )

// y por último construimos el mesh
this.corazon = new THREE.Mesh (geometriaCorazon, suMaterial );
```

Pero no se quiere una figura estática, sino que gire sobre sí misma por el eje Y. Por tanto le ponemos un nodo por arriba que tenga esa transformación. Nos sale el nodo B de la imagen.

Nos vamos a saltar el nodo C por ahora.

Sigamos con el razonamiento.

Queremos que el corazón también gire con respecto a un centro que está fuera del corazón una determinada distancia (dist), pues primero habrá que separarlo del origen esa determinada distancia, y después hacerlo girar con respecto al eje Z.

Nos salen los nodos D (que por ahora hay que verlo como que está conectado directamente con B) y el nodo E.

Pero si lo implementamos y lo ejecutamos vemos que el corazón hace los giros esperados más o menos bien, pero no mantiene la verticalidad. Ello se debe a que ese es precisamente el efecto del nodo E y su rotación por Z. Es como si cogemos un vaso de agua con nuestro brazo horizontal, y ahora manteniendo el brazo recto y sin hacer ninguna articulación en la muñeca, levantamos el brazo por encima de nuestra cabeza, acabamos mojados.

Si no queremos mojarnos ¿qué es lo que hacemos?, según vamos levantando el brazo, vamos articulando la muñeca para que el vaso siempre esté vertical y así no nos mojamos.

Pues con el ejercicio del corazón igual. Ahí es donde aparece el nodo C en el ejercicio, que como véis, cuando el corazón está todavía en el origen, se le aplica la misma rotación en Z que después se le va a aplicar en el nodo B pero con el ángulo opuesto (en negativo), así por cada grado que elevamos el brazo, corregimos dicho grado con la muñeca. El vaso se mantiene siempre vertical y no nos mojamos.

Después el corazón con el ángulo corregido (nodo C) ya se puede aplicar el desplazamiento (nodo D) y lo demás.

Ahora en código, cómo se haría?

Lo primero es ver si ese grafo se puede compactar un poco. Es decir, buscamos cadenas  $T \cdot R_x \cdot R_y \cdot R_z \cdot S$

El Mesh del corazon (nodo A) se puede compactar con el nodo B y su  $R_y$  ( $R_y \cdot \text{mesh}$ )

También vamos a poder compactar los nodos D y C, ( $T \cdot R_z$ ).

Y ya pasamos a codificar.

```
// En el constructor
// Se van construyendo nodos y se van colgando los hijos de los padres
// Si hay alguna transformaci n que no var e en el tiempo , se deja ya hecha .

this.ab = new THREE.Mesh ( la geometria del corazon y su material );

this.cd = new THREE.Object3D ();
this.cd.position.x = dist;    // Esta transformacion es fija , la hacemos en el constructor
this.cd.add (this.ab);

this.e = new THREE.Object3D ();
this.e.add (this.cd);

// En el update
this.e.rotation.z += beta;
this.cd.rotation.z -= beta;
this.ab.rotation.y += alfa
```