

# FAQ

## Sistemas Gráficos

### Grado en Ingeniería Informática

Curso 2019/2020

#### **Pregunta**

¿Cómo hacer el ejercicio práctico 10.c?

#### **Respuesta**

Para seguir la explicación tenéis que tener presente la imagen de la página siguiente.

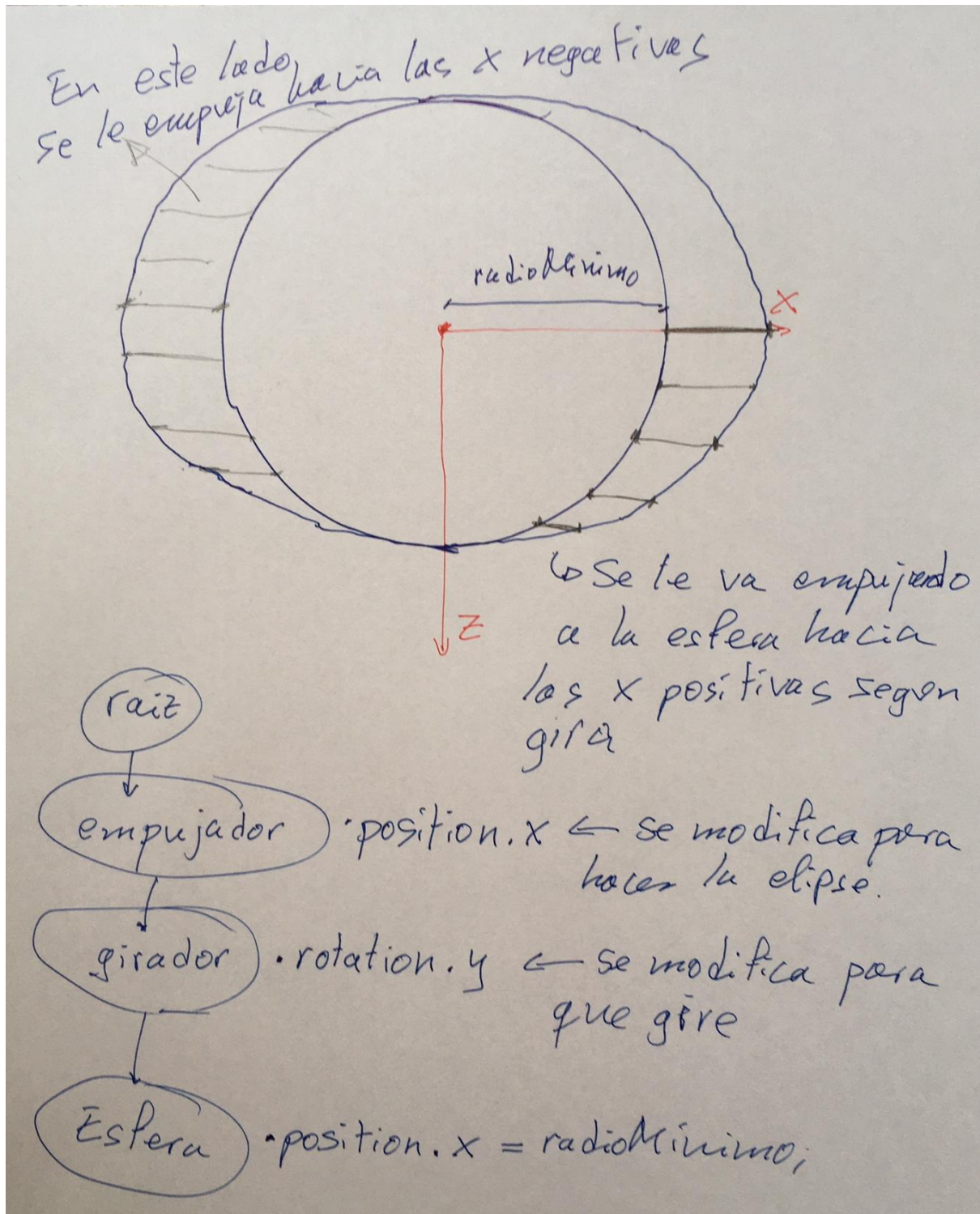
Hacer que una esfera gire alrededor de un punto externo es fácil, ya lo habéis hecho en el ejercicio 10.a y 10.b. ¿Verdad?

No es más que una esfera a la que se le ha aplicado una traslación para alejarla de ese centro y, por encima, se le pone un nodo “girador” con el que gestionamos la rotación.

¿La clave de este ejercicio dónde está? Pues en ponerle por encima un nodo “empujador” para que, al mismo tiempo que la esfera describe ese movimiento circular, se altere su trayectoria.

En el ejercicio 10.b, solo había que ir modificando la `.position.y` del nodo empujador y así la esfera, mientras gira, va subiendo y bajando.

Pues en el ejercicio 10.c es lo mismo, solo que lo que se modifica es la `.position.x` del nodo empujador.



Mirando el dibujo se ve que cuando la esfera esta en la parte inferior del círculo no se le “empuja nada” pero según va avanzando en sentido contrario a las agujas del reloj, cada vez se le “empuja” un poco más (las líneas a lápiz) hasta que llega a un máximo cuando la esfera va

por la  $x$  máxima. Entonces se le va reduciendo el “empuje” hasta que llega la esfera a la parte superior que el “empuje” horizontal es cero.

La esfera sigue en su girar, y el empujador empieza a empujar otra vez pero hacia la parte negativa del eje horizontal hasta que llega al valor -maximo que empieza a reducir el empuje hasta cero y vuelta a empezar.

Este ejercicio se hace mejor con Tween.

Hacen falta dos animaciones, una que modifica el ángulo de giro entre 0 y  $2\pi$  en 4 segundos.

Y otra que modifique el empuje entre +1 y -1 con efecto yoyo, pero de 2 segundos (la mitad del tiempo)

La variable que se modifica con la interfaz gráfica de usuario es la diferencia entre el radio mayor y el menor de la elipse, que llamaremos “desfase”

Cuando el desfase es cero, debe salir un recorrido circular porque los dos radios son iguales.

En el onUpdate de la animación que controla el empujador lo que pondremos será

```
nodoEmpujador.position.x = variableInicioAnimacion.x * that.guiControls.desfase;
```

Comenzando con la esfera en la posición de  $0^\circ$  en el círculo. En cada vuelta completa de 4 segundos, realiza una ida de 2 segundos y una vuelta de otros 2 segundos.

Pues durante la ida, la esfera se ha visto empujada desde  $+1*\text{desfase}$  hasta  $-1*\text{desfase}$  consiguiendo lo que se busca. Y a la vuelta, pues al revés, por el efecto yoyo.

Tan solo hay que añadir a la animación del empujador (a la del giro no) la configuración .easing (TWEEN.Easing.Quadratic.InOut) ya que la velocidad a la que debe ir variando el empuje debe ser más lenta en los extremos y más rápida en el centro. En los extremos la diferencia entre dos empujes consecutivos es mucho menor que la diferencia entre dos empujes consecutivos en el centro. En las líneas a lápiz que representan los empujes en el dibujo se aprecia esto.