

# Ejercicios en Ordenador

Sistemas Gráficos

Grado en Ingeniería Informática

Curso 2021/2022

## Introducción

La siguiente relación de ejercicios tiene como objetivo que el alumnado practique los diferentes conceptos vistos en clase, implementándolos con la biblioteca THREE.JS.

Los ejercicios pueden hacerse individualmente o en grupo. El alumno puede buscar ayuda en otros compañeros, en la documentación de las bibliotecas usadas, en la bibliografía relacionada y, por supuesto, puede consultar sus dudas con el profesor.

Los ejercicios no tienen que entregarse ni serán evaluados. No obstante, el alumno tiene que afrontarlos con el objetivo de aprender a diseñar e implementar soluciones a problemas que requieran usar los conceptos, métodos y técnicas usados en estos ejercicios.

En la fecha que se indique con la suficiente antelación, el alumno será evaluado mediante un examen en ordenador que implicará usar varios de los conceptos, métodos y técnicas vistos en estos ejercicios.

**En la convocatoria se indicará expresamente la materia objeto del examen.** Solo entrará en el examen de prácticas los conceptos que hayan sido vistos en clase de teoría.

## Ejercicios

### 1. Hola mundo Three.js

Este ejercicio muestra una aplicación mínima usando THREE.JS para la programación gráfica y la biblioteca DAT.GUI.JS para la interfaz gráfica de usuario (GUI).

Es el único ejercicio en el que se proporciona el código fuente. El alumno debe estudiar la aplicación y comprender su diseño.

### 2. Geometría básica 3D

Mediante este ejercicio el alumno debe familiarizarse con las diferentes figuras 3D que proporciona la biblioteca THREE.JS y conocer sus principales parámetros.

El vídeo `geometria-basica.mp4` muestra diversos ejemplos de varias figuras donde se muestran los efectos de dar diferentes valores a sus parámetros.

No es necesario que el ejercicio sea como lo que se muestra en el vídeo.

**No es necesario que se puedan modificar las figuras interactivamente. Basta con que se muestren diferentes figuras con diferentes valores en sus parámetros.**

*A tener en cuenta*

- Modificar una geometría ya creada implica volver a crearla. Debe evitarse crear nuevas geometrías para cada frame, ya que ralentiza mucho la visualización. En cualquier caso deben destruirse (con `dispose`) las geometrías desechadas.
- El material usado en el vídeo es `MeshNormalMaterial`, que asigna los colores a los polígonos según el vector normal de sus caras o sus vértices.  
El sombreado plano o suave se consigue asignándole `true` o `false`, respectivamente, al atributo `flatShading` del material. Tras modificar dicho atributo hay que asignar `true` al atributo `needsUpdate` del material para que el cambio sea tenido en cuenta en el siguiente frame.
- El movimiento continuo se consigue incrementando un poco la rotación de cada figura en cada frame. Por ejemplo: `this.caja.rotation.y += 0.01;`

### 3. Geometría por Revolución

Mediante este ejercicio el alumno aprenderá a crear geometría por revolución a partir de una línea definida en un plano.

El vídeo `geometria-revolucion.mp4` muestra un posible resultado del ejercicio.

**No es necesario que se puedan modificar las figuras interactivamente. Basta con que se muestren diferentes figuras con diferentes valores en sus parámetros.**

*A tener en cuenta*

- La línea a revolucionar se define como un array de `Vector3` con la  $z = 0$  y se usa dicho array, tal cual, al crear la geometría por revolución. Sin embargo, para crear y visualizar una línea, se deben añadir esos puntos a una `BufferGeometry` y después usar esa `BufferGeometry` al instanciar la clase `Line`.

**Ejemplo:**

```
// Puntos
var points = [];
// Se añaden puntos al array
//   mediante unas cuantas instrucciones como la siguiente
//   definimos el contorno a revolucionar
points.push (new THREE.Vector3 (unX, unY, 0));

// Para crear la figura por revolución
var latheObject = new THREE.Mesh (
    new THREE.LatheGeometry (points, ...), unMaterial);
// Para crear una línea visible, como en el vídeo
var lineGeometry = new THREE.BufferGeometry();
lineGeometry.setFromPoints (points);
line = new THREE.Line (lineGeometry, unMaterial);
```

- Los ángulos se dan en radianes.

**4. Geometría por Barrido**

Mediante este ejercicio el alumno aprenderá varias cosas.

- a) Crear contornos con `THREE.Shape`.
- b) Usar dichos contornos para realizar extrusiones (con y sin bisel).
- c) Usar dichos contornos para realizar barridos por una trayectoria definida mediante puntos.

El vídeo `geometria-barrido.mp4` muestra un posible resultado. En el vídeo, las figuras diamante y corazón están hechas con extrusión con bisel. Las figuras pica y trébol están hechas también con extrusión con bisel salvo el pie que está hecho con una revolución. Para las columnas se han aprovechado los contornos del trebol y corazón para hacer un barrido por una trayectoria de varios puntos.

**No es necesario implementar las rotaciones que se muestran en el vídeo.**

**Basta con que se aprenda a crear figuras con esa topología: el barrido de un contorno.**

*A tener en cuenta*

- Para la trayectoria del barrido se ha usado `THREE.CatmullRomCurve3`.

### 5. Geometría de Sólidos Constructiva (CSG)

Con este ejercicio el alumno se familiarizará con las operaciones booleanas como medio de construir sólidos con geometría compleja a partir de otros sólidos más sencillos.

El vídeo `geometria-solidos-constructiva.mp4` muestra el resultado.

**No es necesario implementar las rotaciones que se muestran en el vídeo. Ni realizar exactamente todas las figuras mostradas.**

**Basta con que se aprenda a crear figuras con esta técnica.**

**Eso sí, para un mejor aprendizaje es IMPORTANTE fijarse una figura objetivo e intentar reproducirla. No realizar operaciones booleanas *sin ningún sentido* a ver lo que sale.**

*A tener en cuenta*

- En cada operación se puede partir de sólidos que se hayan generado a partir de:
  - Primitivas
  - Revolución
  - Barridos
  - Otras operaciones booleanas
- La rosca de la tuerca, que supone bastantes operaciones entre los triángulos de las mallas de polígonos correspondientes, tarda un cierto tiempo en completarse.

### 6. Cargar un modelo en formato .obj

Este ejercicio consiste en cargar un modelo en formato .obj. Se proporciona un modelo, pero el alumno puede buscar otro si lo desea.

Se muestra un posible resultado en el vídeo `modeloCargado.mp4`.

## 7. Péndulos

Este ejercicio consiste en diseñar correctamente e implementar un modelo jerárquico que se corresponda con los péndulos mostrados en el vídeo `pendulos.mp4`.

**Es primordial realizar correctamente este tipo de ejercicios.**

Este en concreto es bastante completo pues combina traslaciones, rotaciones y escalados con dependencias entre dichos grados de libertad.

### *Explicación de los grados de libertad del ejercicio*

- La figura contiene 2 péndulos que oscilan con respecto a sus respectivos ejes.
- El péndulo superior contiene una parte central roja cuyo tamaño en Y es variable entre 5 y 10 unidades mediante un escalado de dicha parte. **Grado de libertad:** Longitud del péndulo superior.
- En los extremos de esta parte roja hay dos partes verdes, con un tamaño en Y fijo de 4 unidades. Estas partes verdes nunca se separan ni se intersecan con la roja aunque la roja varíe de tamaño.
- El eje del péndulo superior está situado a 2 unidades en Y desde su parte superior. Es decir, está centrado con la parte verde superior.
- El péndulo oscila a un lado y otro por su eje un máximo de  $45^\circ$  en cada dirección. Es decir, entre sus 2 extremos hay un ángulo de  $90^\circ$ . **Grado de libertad:** Oscilación del péndulo superior.
- El eje del que oscila el péndulo inferior se desplaza por la parte roja del péndulo superior.
- Ese desplazamiento se produce entre el 10% y el 90% de la longitud de la parte roja del péndulo superior. Sea cual sea la longitud de dicha parte roja. **Grado de libertad:** Posición del eje del péndulo inferior.
- Independientemente, el péndulo inferior también puede alargar su tamaño en Y entre 10 y 20 unidades. **Grado de libertad:** Longitud del péndulo inferior.
- El eje del péndulo inferior está siempre a 1 unidad en Y desde su parte superior. Con independencia de su longitud.
- El péndulo inferior también puede oscilar hasta un máximo de  $45^\circ$  en cada dirección con respecto a la posición del péndulo superior. Es decir, ese ángulo es el que forman los 2 péndulos. **Grado de libertad:** Oscilación del péndulo inferior.

### *A tener en cuenta*

- En este tipo de ejercicios, un buen diseño del grafo es fundamental. Se debe realizar primero el diseño del modelo jerárquico realizando su correspondiente grafo y después pasar a la implementación.
- El alumno que lo desee puede entregar su diseño al profesor para que sea corregido antes de pasar a la implementación.

### 8. Animación con control de velocidad

Realizar una especie de reloj como el mostrado en el vídeo `reloj.mp4`. Con el deslizador de la interfaz gráfica de usuario se controla la **velocidad**, expresada en marcas alcanzadas por segundo. La velocidad puede hacerse 0, deteniéndose la esfera móvil, o puede ser negativa, con lo que el móvil gira en sentido contrario.

**Importante:** En este ejercicio se persigue un control preciso de la velocidad de movimiento. El control de la interfaz debe mostrar un número que indica la velocidad del móvil en marcas/segundo. Debe cumplir dicha velocidad, y debe hacerlo en cualquier ordenador sin tener que modificar el código de un ordenador a otro.

### 9. Recorridos

El vídeo `recorridos.mp4` muestra un camino en forma de 8, definido mediante un spline. Una nave recorre dicho camino cíclicamente con los siguientes parámetros:

- En recorrer el bucle de la derecha emplea 4 segundos.
- En recorrer el bucle de la izquierda emplea 8 segundos.
- Cada bucle lo recorre empezando lento y acabando lento, por eso se aprecia una pequeña parada cada vez que pasa por el eje Y del sistema de coordenadas.

Diseñar e implementar (usando Tween) una aplicación similar a esta.

Se trata de un ejercicio bastante completo ya que supone definir un movimiento como una composición de 2 animaciones consecutivas, cada una de ellas con un principio, un final y un tiempo empleado. Además de tener que programar la velocidad para que el recorrido de cada bucle lo haga lento al salir y lento al llegar.

### 10. Interacción

En este ejercicio (que se muestra en el vídeo `interaccion.mp4`) se ponen en práctica varios conceptos relacionados con la interacción del usuario con la aplicación:

- Se muestran mensajes en pantalla
- Se añaden botones a la interfaz de usuario
- Se muestra un pop-up ante la pulsación de una tecla
- Se leen y procesan eventos del ratón, pulsación de los botones, movimiento de la rueda y arrastre del ratón.
- Los eventos del ratón manipula la cámara o las cajas según se tenga pulsada la tecla Ctrl o no, respectivamente.
- La caja que se ha seleccionado se pone parcialmente transparente mientras se opera con ella.

*De manera opcional*

El vídeo también muestra una detección de colisiones sencilla: Puede considerarse que 2 cajas colisionan si la distancia entre sus centros es menor que una determinada distancia. La caja que está en movimiento, si colisiona con otras cajas, se coloca encima. Además, si intentamos mover una caja que está debajo de otras, las de arriba caen.

*A tener en cuenta*

Lo que muestra el vídeo es una parte del ejemplo de la grúa que se ha dado en clase y cuyo código está en Prado.

Al igual que en ejercicios anteriores, **no es necesario realizarlo exactamente igual, pero sí es importante que se aprenda a añadir figuras en un escenario haciendo clic con el ratón y a seleccionar figuras con el ratón para hacerles algún tipo de modificación.**

**11. Órdenes por teclado**

Mediante este ejercicio se capturarán los eventos relacionados con pulsar y soltar una tecla y se responderá ante ellos. En el vídeo `teclado.mp4` se muestra un posible resultado. El coche avanza y retrocede como respuesta a los cursores arriba y abajo, respectivamente, y gira a izquierda y derecha como respuesta a los cursores izquierda y derecha respectivamente. Se pueden pulsar varias teclas a la vez y el coche responde a ellas.

**12. Vistas**

Mostrar una escena utilizando varios tipos de vistas en varios viewports. En el vídeo de ejemplo, `vistas.mp4` se muestra el navegador dividido en 4 viewports. En el superior izquierdo se muestra una vista en perspectiva que se puede modificar con el ratón (se ha usado `TrackballControls`). En la vista inferior izquierda se muestra una vista subjetiva desde el punto de vista del conductor del coche. En el lado derecho hay 2 vistas paralelas: una vista en planta en el lado superior y una isométrica en el lado inferior ([https://es.wikipedia.org/wiki/Proyección\\_isométrica](https://es.wikipedia.org/wiki/Proyección_isométrica)).

**13. Luces**

En este ejercicio se trabaja con diferentes tipos de luces, en concreto:

- Luz ambiental: Que permite que siempre haya un mínimo de luz en la escena, no quedando nada completamente negro.
- Luz direccional: Usada para simular la luz del sol. Los rayos llegan paralelos entre sí. En el ejemplo del vídeo se ha configurado para que genere sombras.
- Luz puntual: Usada para representar la luz de la farola. Los rayos parten de la farola hacia todas las direcciones. También se ha configurado para que genere sombras.
- Luz focal (spot): Usada para representar los faros del coche. La luz parte de las bombillas del coche con una determinada dirección y un determinado ángulo de apertura. Estas luces, además, son subjetivas. Se mueven solidariamente con el coche.

Además, se ha usado la componente emisiva del material de la luminaria de la farola y de las bombillas del coche para simular que son estas fuentes las que emiten la luz, viendose más intensas cuando están encendidas.

El vídeo `luces.mp4` muestra un posible resultado del ejercicio.

**No es necesario que se haga exactamente igual, pero hacer un ejercicio donde se usen diferentes tipos de luz.**

#### 14. Texturas

Este ejercicio consiste en crear una escena con objetos sencillos y aplicarle materiales con texturas en el canal difuso. En el ejercicio se deben aplicar las texturas con diferentes modos. Con repetición y sin repetición, con repetición en espejo y sin espejo. Giradas y desfasadas, de manera que se practiquen los distintos modos de aplicación de las texturas. El vídeo `texturas-canal-difuso.mp4` muestra diferentes modos de aplicación de un par de texturas a un plano cuadrado.

**No es necesario que vuestro ejercicio sea exactamente igual que éste, ni que sea interactivo. Basta con que haya varios objetos y a cada uno se le apliquen las texturas de una manera diferente.**

#### 15. Materiales

Mediante este ejercicio tenéis que crear distintos materiales que usen diferentes colores, con brillo y sin él, así como materiales que usen diferentes tipos de texturas en distintos canales.

El vídeo `materiales.mp4` muestra un ejemplo.

En la escena del vídeo hay 4 cajas que, de izquierda a derecha, muestra un material con solo textura difusa, un material con textura difusa y de relieve “bump”, otro con textura difusa y una textura de ajedrez en el canal alpha que hace que el cubo se vea recortado y un cubo donde se mezclan todas las anteriores: color, relieve y recortes.

En la parte superior hay 4 esferas, las 2 de la izquierda tienen un color dorado pero una de ellas tiene brillo mientras que la otra es mate. Las 2 esferas de la derecha muestran una textura de mármol con brillo; pero en una de ellas la textura se ha aplicado tal cual, con los colores blancos originales de la imagen, y en la otra se ha aplicado combinándola con un color dorado, consiguiendo con la misma imagen un material distinto al combinar la imagen con un color.

En medio de las esferas hay un plano cuyo material muestra como textura un vídeo que se está reproduciendo.

**Una vez más, el ejercicio consiste en que practiquéis con distintos materiales que usen texturas en diferentes canales. No tiene por qué ser igual al del vídeo.**