

Práctica Final – Pong en Three.js

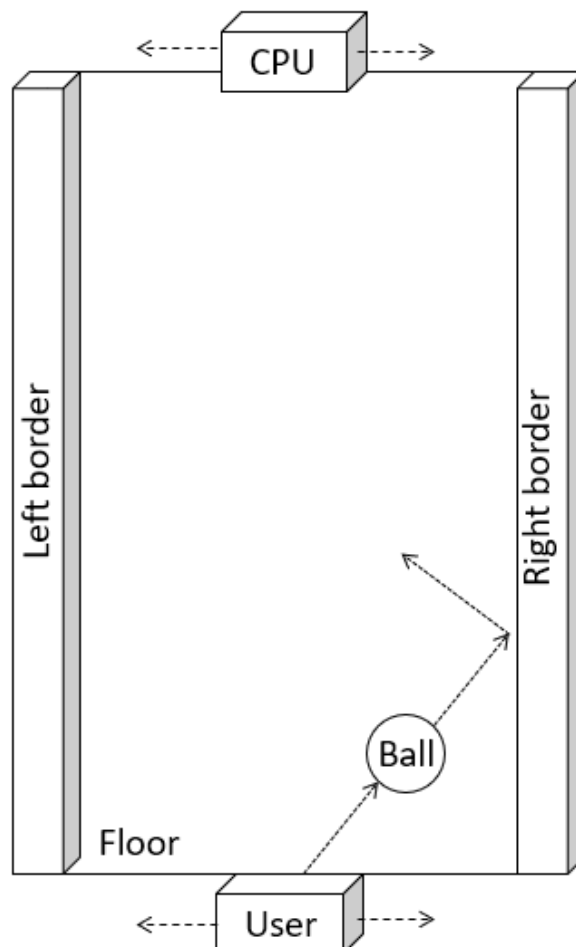
El objetivo de esta práctica es desarrollar una aplicación web que contenga un gráfico 3D interactivo creado con Three.js. Este gráfico implementará el juego clásico: [Pong](#).

Puntos totales posibles del ejercicio: 10

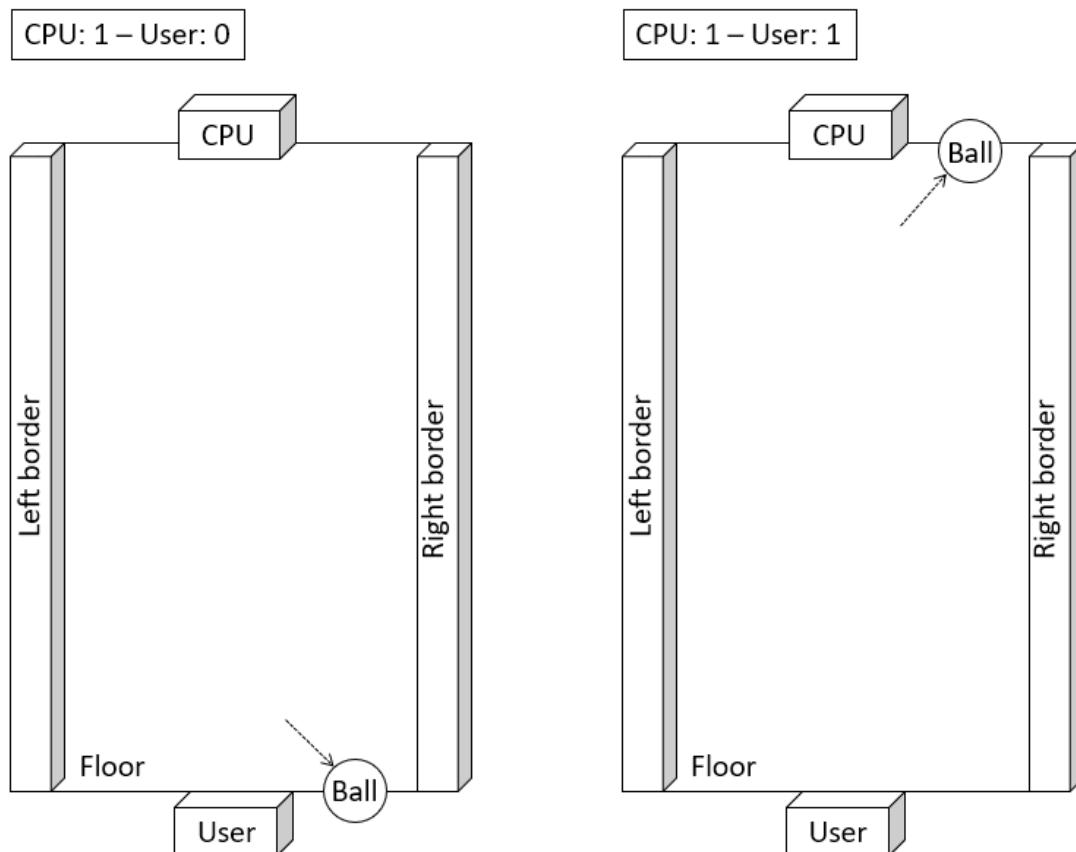
Funcionamiento básico

El gráfico 3D que implemente el juego estará formado por los siguientes elementos:

1. Suelo. Superficie plana en 2D por la que se desplazará la bola.
2. Bola. Esfera en 3D que se utilizará en el desarrollo del juego.
3. Bordes izquierdo y derecho. Hexaedros en los laterales del plano de juego (suelo).
4. CPU. Hexaedro situado en la parte superior del plano. Su movimiento en el eje X será automático.
5. Usuario. Hexaedro situado en la parte inferior del plano. Su movimiento en el eje X será realizado por el jugador, capturando las pulsaciones de teclado de las flechas izquierda y derecha.



La bola se desplazará por encima del suelo variando la posición de sus coordenadas X e Y. Existirá detección de colisión de la bola con respecto a los bordes derecho e izquierdo, así como con respecto a los hexaedros de CPU y usuario. El juego consistirá en devolver la bola por parte de la CPU y el usuario, desplazando los respectivos hexaedros hasta bloquear el paso de la bola en el momento en el está en la parte superior e inferior del plano respectivamente. Si el hexaedro de CPU/usuario bloquean esta posición, la bola rebotará en la dirección contraria. En caso contrario, la bola saldrá del plano y se añadirá un punto al jugador (CPU o usuario) que haya conseguido hacer caer la bola por el lado contrario. Por ejemplo:



En algún lugar de la página web deberá existir un marcador que muestre la puntuación. En el funcionamiento básico, no es necesario que sea dentro del gráfico Three.js (se puede hacer con elementos HTML de tipo texto). Se empezará con 0-0 y el primer jugador en llegar a 5 puntos será el vencedor.

Al cargar la página el juego no empezará hasta que el usuario pulse una tecla (por ejemplo, la barra espaciadora). En ese momento la bola empezará a moverse en una de las 2 direcciones del plano (hacia CPU o usuario). El hexaedro de la CPU se moverá automáticamente siguiendo la trayectoria de la coordenada X de la esfera, mientras que el hexaedro del usuario se moverá la izquierda o derecha en función de la pulsación de las teclas \leftarrow y \rightarrow del teclado. Cuando CPU o usuario marquen un punto el juego se parará y se actualizará el marcador. Al llegar a 5 se declarará ganador a uno de los jugadores y se ofrecerá la opción de volver a jugar.

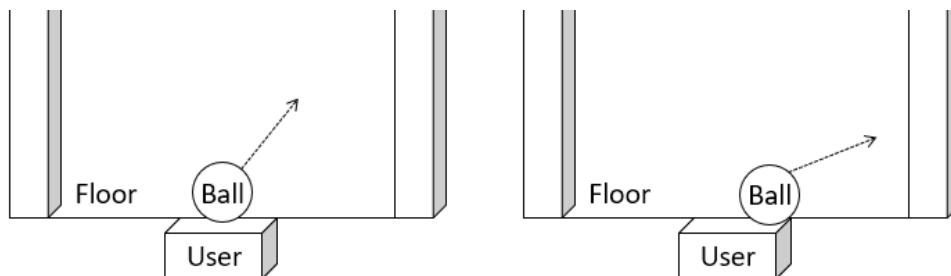
El modo de funcionamiento básico supone que se usarán los siguientes elementos:

- Texturas para todos los objetos (suelo, bordes, CPU, usuario, y bola).
- Fuente de luz direccional o puntual. Además, La bola deberá producir efectos de sombra, que será reflejada tanto en el suelo, como en los bordes y hexaedros (CPU/usuario).

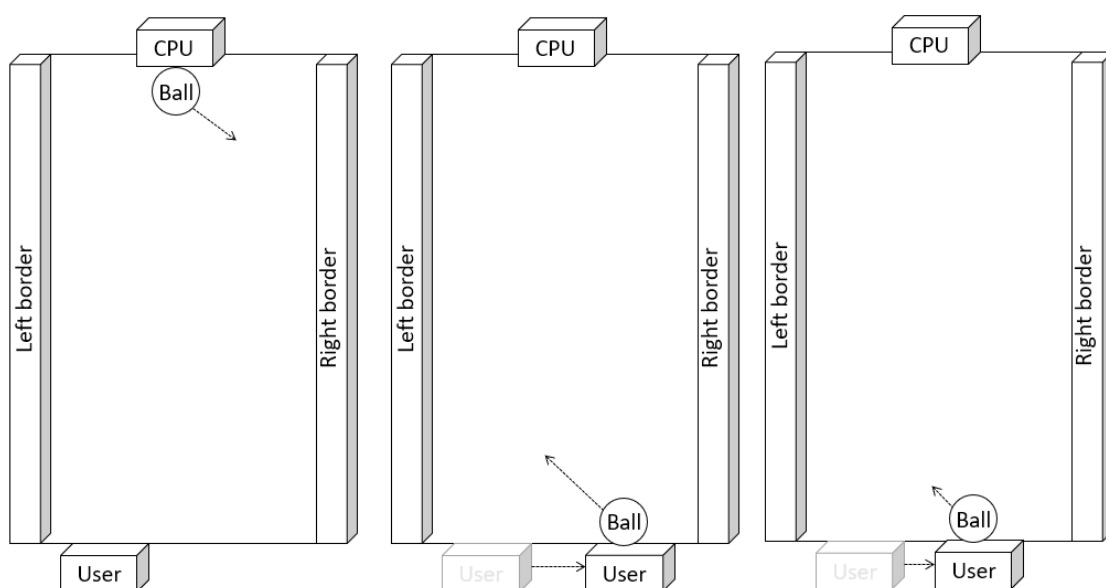
Funcionamiento avanzado

En el funcionamiento básico se supone que la velocidad con la que la bola se desplaza verticalmente (esto es, como se mueve a lo largo del eje Y) y el ángulo con el que rebota en los bordes y jugadores (esto es, como se mueve a lo largo eje X) es constante. En el funcionamiento avanzado estos 2 parámetros serán variables.

Para variar el **ángulo de rebote**, se tendrá en cuenta la posición del hexaedro con respecto a la bola a la hora de bloquearla. El ángulo será mayor si se bloquea con el extremo del hexaedro, y menor si se bloquea de forma centrada.



Para la **velocidad de desplazamiento vertical** se tendrá en cuenta el desplazamiento del hexaedro de su posición inicial (cuando el jugador contrario rebota la pelota) respecto al punto donde se bloquea. A mayor desplazamiento horizontal, mayor velocidad de desplazamiento vertical.



Tanto en el modo básico como en el modo avanzado se da por supuesto que la CPU va a poder bloquear siempre la bola. Para que sea un juego más realista, debería existir la **posibilidad de poder ganar a la CPU**. Para ello, habría que configurar adecuadamente la velocidad con la que la CPU se desplaza horizontalmente, de modo que existan situaciones en las que no le dé tiempo a llegar a bloquear la bola. Se puede incluso introducir una componente de aleatoriedad en el desplazamiento del hexaedro de la CPU para conseguir este efecto.

Mejoras

Se deja abierto una parte de calificación (ver sección evaluación) para introducir mejoras extras en la práctica. Las posibilidades de mejora pueden ser variadas. Se calificarán las mejoras que mejoren la experiencia de uso de la aplicación. Algunas ideas son:

- Introducir niveles de dificultad. Pueden ser configurable al iniciar el juego o ir aumentando según se juegue una y otra vez.
- Marcador con gráfico 3D en lugar de como texto HTML.
- Posibilidad de elegir diferentes texturas para los elementos de juego (“skin”).
- Animaciones adicionales (movimiento de la cámara o punto de luz en función del juego).
- ...

Para optar a la nota máxima (10 puntos) habrá que implementar al menos dos mejoras extras.

Evaluación

Esta práctica se evaluará con un máximo de 10 puntos. La distribución de estos puntos será de la siguiente manera:

- Funcionamiento básico: 5 puntos.
- Funcionamiento avanzado: 3 puntos.
- Mejoras: 2 puntos.

Además, se tendrá en cuenta la calidad de código. Se supone que el código deberá cumplir los siguientes principios:

1. Se deben evitar los fragmentos de código duplicados (principio DRY, Don’t Repeat Yourself)
2. El código debe ser legible. Esto se traduce en:
 - a. Usar nombres de variables, funciones, etc. significativos (preferiblemente en inglés).
 - b. El código debe estar correctamente indentado (usando tabs o espacios).
 - c. No se utilizará código innecesario (por ejemplo, funciones que no se usan, condiciones que no se cumplen, etc).

Si se detectan problemas de calidad de código en base a estos dos principios, se restará hasta 1 punto de la nota obtenida al sumar las componentes anteriores (funcionamiento básico, avanzado, y mejoras).

Consejos

Es aconsejable seguir las siguientes recomendaciones:

- Comenzar implementando la funcionalidad básica. Una vez realizada, si se aspira a sacar más de 5 puntos, se puede implementar la funcionalidad avanzada y por último las mejoras extras.
- Ir paso a paso. La mejor estrategia sería Implementar pequeños cambios comprobando verificando en cada momento que todo funciona como se espera.
- Usar la consola de depuración del navegador (Chrome, Firefox) para visualizar las trazas en la consola JavaScript (console.log) así como los posibles errores según vamos desarrollando.

Entrega

La entrega se realizará de forma **individual**. Hay que subir un archivo comprimido .zip en el aula virtual con todos los ficheros de la práctica, esto es:

- Página web (fuente HTML).
- Imagen(es) de textura.
- Librerías JavaScript. Si se usa código JavaScript fuera de la página HTML (aunque no es necesario que el código JavaScript vaya en un fichero aparte, puede ir embebido dentro de la página web).

La práctica será evaluada en primer lugar ejecutada desde un navegador Chrome y servida con un servidor web local.

En la cabecera de la página web deberá aparecer un comentario (esto es, un párrafo entre los símbolos `<!--` y `-->`) informando de las partes que has implementado. Por ejemplo:

```
<!--  
  
Partes implementadas:  
  
- Funcionalidad básica  
- Funcionalidad avanzada (ángulo de rebote variable, velocidad de  
desplazamiento vertical variable, posibilidad de ganar a la CPU)  
  
-->  
<!DOCTYPE html>  
<html>  
  
<head>  
<title>Pong</title>
```

El ejemplo anterior aspiraría a sacar una máxima de 8 puntos (5 de la funcionalidad básica y 3 de la funcionalidad avanzada).