El objetivo del ejercicio es implementar un sistema de temporización base para un videojuego. Nos basaremos en la teoría mostrada en las transparencias que se encuentran en Tema 3. Temporización.

La idea es modificar la solución del ejercicio anterior para introducir el sistema de control del tiempo.

**Ejercicio 2.1**

El programa actual muestra unas bolas colisionando entre ellas. El control de tiempo se ha realizado de manera artificial utilizando SYS\_Sleep() que se encarga de esperar 17 ms en cada iteración y así forzar los 60 fps aproximados. Pero no es una solución que independice de la velocidad de proceso de la máquina.

Quitar el V-Sync del driver de video.

Comentar la función SYS\_Sleep del código.

Ejecutar el nuevo código y compararlo con la ejecución de la versión anterior, para comprobar más claramente las consecuencias de no tener un sistema de control de tiempo.

* Al comentar la función Sleep(17), la velocidad de las pelotas en ejecución aumenta, ya que se actualiza su posición 1 vez por frame sin tener en cuenta el tiempo de actualización.

**Ejercicio 2.2.**

Utilizar las soluciones dadas en las transparencias para tener un sistema de control de tiempo para la lógica del juego, sin implementar la solución a “la espiral de la muerte”.

Usar las instrucciones QueryPerformanceFrequency y QueryPerformanceCounter para la obtención del tiempo actual.

Hacer que la frecuencia de actualización de la lógica sea 60 fps.

* Hecho

**Ejercicio 2.3.**

Es necesario mostrar información en pantalla del estado de los diferentes contadores de tiempo para comprobar que efectivamente el control de los tiempos está funcionando correctamente.

Mostrar en el FPS (Frames per second). Es el número de veces que se ha llamado a la función de Render por segundo.

Mostrar el tiempo real transcurrido desde el comienzo del juego.

Mostrar el tiempo que ha transcurrido ejecutándose la lógica desde el comienzo del juego.

* FPS Hecho.
* Real time Hecho
* Logic Time Hecho

**Ejercicio 2.4.**

Comprobar que la velocidad de las bolas sigue siendo la misma, aunque se descomente la función SYS\_Sleep, con lo que se ha creado un sistema de control básico del tiempo independiente de la plataforma en la que se ejecute.

* Sigue siendo la misma

**Ejercicio 2.5.**

Comprobar que el problema de “la espiral de la muerte” no está solucionado.

Para ello pondremos dentro del bucle de lógica la instrucción SYS\_Sleep(17), teniendo presente que la frecuencia de actualización de la lógica es de 60 fps.

Comprobar que el FPS baja progresivamente ya que se ha simulado un tiempo de proceso de la lógica que está por encima de los 60 fps.

* Efectivamente cada vez es más lento

**Ejercicio 2.6.**

Introducir solución para evitar la espiral de la muerte, haciendo que el tiempo acumulado de lógica nunca supere 1 / 15 sec.

Comprobar que el frame rate no disminuye progresivamente y se mantiene estable.

Comprobar que la diferencia entre el tiempo de lógica y el real es cada vez mayor.

**Ejercicio 2.7.**

Introducir que la velocidad de las bolas dependa del tiempo.

Variar la simulación de tiempo del mundo de modo que vaya más deprisa o más rápido que el tiempo real. Para ello haremos que el tiempo de la simulación sea un factor del tiempo real.

Comprobar que el paso del tiempo real y tiempo de procesado de mundo son realmente diferentes.