



PROGRAMACIÓN DE SISTEMAS Y DISPOSITIVOS

PROYECTO FINAL

Fecha de entrega, hasta el 18 de enero de 2026 a las 23:59h

Space Invaders es un videojuego desarrollado originalmente para máquinas recreativas por la compañía japonesa Taito en 1978. El siguiente [enlace](#) contiene un *gameplay* del videojuego, en el que se observa cómo el jugador controla una nave espacial para defender la Tierra de enjambres de invasores extraterrestres, parapetado tras unos edificios.

El objetivo del proyecto es realizar una adaptación lo más fidedigna posible de *Space Invaders* en la placa de prototipado Embest S3CEV40, haciendo uso de los siguientes recursos:

- *LCD*: para la visualización del juego.
- *Audio codec*: para la reproducción de sonidos.
- *Keypad*: para el control de la nave espacial.
- *Pushbuttons*: para emular la inserción de una moneda.
- *Display* de 7 segmentos: para la visualización del crédito disponible.

La aplicación tendrá una arquitectura multitarea cooperativa bajo un *kernel* de planificación no expropiativo que usa la técnica de *pooling* periódico para la lectura del *keypad* y *pushbuttons*. El *timer0* será la única fuente de interrupción permitida e interrumpirá 100 veces por segundo (tick) que disparará la planificación y despacho de las tareas. Los sonidos se reproducirán por DMA.

Esta aplicación deberá tener, al menos, las siguientes tareas:

- Lectora de *keypad* por *pooling* periódico.
- Lectora de *pushbuttons* por *pooling* periódico.
- Lanzadora del UFO.
- Lanzadora del disparo enemigo.
- Actualizadora de la posición de la nave espacial del jugador.
- Actualizadora de la posición del disparo del jugador.
- Actualizadora de la posición del disparo del enemigo.
- Actualizadora de la posición del UFO.
- Actualizadora de la posición del enjambre.

El código se programará en C, siguiendo una organización modular orientada a objetos, un enfoque muy común en firmware y sistemas empujados. Este estilo permite encapsular datos y funciones asociadas a ellos mediante estructuras y punteros a las mismas, simulando las características de la programación orientada a objetos, como métodos, estado interno y encapsulación, sin necesidad de usar C++ u otros lenguajes con soporte nativo para objetos.

Para facilitar la realización del proyecto final, y en especial a aquellos estudiantes que no hayan completado todos los laboratorios, se suministra un *workspace* configurado que incluye:

- Una biblioteca precompilada de firmware (BSP/Debug/libBSP.a) y un directorio conteniendo los ficheros cabecera de las funciones de la biblioteca (BSP/include). Esta biblioteca implementa todas las funciones de acceso a los distintos dispositivos de la placa de prototipado estudiados durante el curso.

- Un proyecto demo configurado que incluye una tarea que emula la inserción de monedas en la recreativa para aumentar el crédito mediante pulsación de cualquier *pushbutton*. El crédito disponible se visualiza en el *display* 7-segmentos.
- Una configuración de depuración que permite la carga y prueba del programa demo en la placa de prototipado.

Los estudiantes que hayan finalizado todos los laboratorios podrán elegir entre usar este *workspace* o el *workspace* del curso con su propia biblioteca.

Adjunta se pueden encontrar la colección de transparencias usadas en clase en donde se explican detalladamente algunos aspectos de la aplicación.