

PROYECTO HARDWARE

RESUMEN EJECUTIVO

Diego Santolaya Martínez (727209)
Abel Chils Trabanco (718997)

El proyecto está basado en las prácticas 1, 2, y 3 de la asignatura proyecto hardware, durante las que partiendo del código del juego del reversi se ha conseguido hacerlo compatible en una placa del desarrollo con arquitectura ARM. Esta compatibilidad se traduce en poder jugar al reversi en las placas controlándolo mediante la pantalla táctil y los botones físicos de los que dispone la placa y visualizando el estado del juego tanto en el 8led como en la pantalla de esta.

Para lograr estos objetivos se han tenido que realizar bibliotecas que interactuasen con el hardware, para gestionarlo y lograr una abstracción de este. Por un lado, para gestionar los botones se desarrolló una en la cual además de comunicarse con el hardware, gestiona los cambios que se producen en la señal que produce el botón cuando este se pulsa debido a circunstancias físicas. También se desarrolló otra biblioteca para realizar la gestión de la pantalla táctil, encargándose tanto de realizar la interacción con el hardware para descubrir pulsaciones de esta, como de calibrarla para producir una experiencia óptima de juego. Por último destacar también otra biblioteca desarrollada que se encarga de interactuar y dibujar sobre la pantalla, en la cual se trabaja tanto a nivel de la memoria virtual de la pantalla como de la memoria real de esta y se interactúa con el controlador de DMA a la hora de realizar la transferencia de información.

Por otro lado se configuró e hizo uso de diferentes timers que se usaron tanto como apoyo para otras bibliotecas como la de gestión de los botones, como para medir tiempos. Para medir tiempos se utilizaron dos timers, el primero configurado para ofrecer la máxima precisión posible para hacer mediciones con alta precisión, estableciéndose las interrupciones que producían este como tipo FIQ y usándose para hacer mediciones de tiempos de ejecución. El otro se configuró con la mínima precisión posible y fue utilizado para medir el tiempo de juego, el cual se mide en segundos.

Para adaptar el juego al hardware, se diseñaron bibliotecas de más alto nivel, que se comunican con las bibliotecas que interactuaban con el hardware y son capaces tanto de detectar regiones de pantalla pulsadas, así como son las encargadas de dibujar todas las pantallas que se utilizan en el juego. Estas bibliotecas son utilizadas desde una máquina de estados principal del programa que se encarga de gestionar la entrada salida utilizando tanto estas como el resto de bibliotecas creadas y simular el juego.

Como apoyo en las labores de desarrollo se desarrolló una pila de depuración, en la cual se pueden guardar eventos y un el momento en el que este se produjo, siendo esta de vital importancia a la hora de depurar eventos que se producen de forma asíncrona, como son las interrupciones.

Debido a que se disponía de un acceso limitado a la placa, se tuvieron que diseñar mecanismos para poder desarrollar el proyecto si ella, entre otros cabe destacar una biblioteca de renderización de imágenes que se desarrolló, la cual posee una interfaz igual a la que posee la biblioteca que interactúa con la pantalla y su resultado es el mismo que se mostraría sobre la pantalla, pero en formato de imagen.

La ejecución del proyecto durante su desarrollo puede provocar múltiples errores, por ello se realizó una gestión de las excepciones que puede producir el procesador, para descubrir si estos se producían.

Por otro lado, el juego no necesita tener un modo privilegiado del procesador para ejecutarse, por ello se decidió hacer funcionar el procesador en modo supervisor lo minimizo posibles, de decir solamente durante la inicialización de todo el sistema y de los periféricos, realizándose el resto de la ejecución del juego en modo usuario.

Para dar más eficiencia al juego, se decidió compilar su versión final con un nivel de optimización o3, lo que causo algunos fallos, producidos en algunos casos por no definir no establecer los atributos de ámbito de las variables como static, const o volatile correctamente, así otros derivados de fallos del compilador, que tuvieron que ser solventados.

Por último de debió de hacer que la placa pudiese funcionar de forma autónoma con el juego, para ello se debieron de enfrentar a dos retos, el primero consistía en copiar el programa a la placa y hacer que esta lo iniciase al encenderse, y el segundo consistía en que el propio programa se copiase a la dirección de memoria RAM, en la que debería de ejecutarse.