



PROYECTO DE PRÁCTICAS

Estructura de Computadores. Juego: Blocky



Juan Perea Campos

José Antonio Martínez López

Francisco López Navarro

Rubén Pérez Rubio

Pablo Moraga Navas

Grupo: G1. Equipo: "Blocky"

19 DE MAYO DE 2014

Manual de usuario:**Instrucciones para jugar:**

- Al encender la consola saldrá una pantalla principal en la pantalla táctil con un mensaje de "PLAY GAME". Para iniciar el juego solo tendremos que pulsar encima de éste.
- Una vez comenzado el juego, el argumento de este trata de llegar hasta el bloque verde, mediante el movimiento del sprite. Dicho movimiento se realizará con las flechas de dirección del Pad.
- Para llegar hasta la zona verde habrá que ir chocando contra los bloques amarillos. Una vez pulsada una tecla de dirección, el sprite no dejará de moverse hasta chocar con uno de los bloques amarillos. Tampoco será posible cambiar la dirección mientras el sprite esté en movimiento.
- Al llegar a la zona verde, te habrás pasado el nivel y te saldrá una pantalla con la opción de pasar al siguiente. Para ello hay que pulsar encima del mensaje "NEXT".
- En caso de chocar contra los límites de la pantalla automáticamente perderás y te saldrá una pantalla para poder reiniciar el nivel. Para esto solo tendrás que pulsar sobre el mensaje "RETRY".
- Tendrás un timer/contador, el cuál irá contando en segundos el progreso del juego.
- El juego consta de 3 niveles, y al terminarlos todos vuelves al inicio tras pulsar en un mensaje que se muestra en pantalla felicitándote por tu hazaña.

Conceptos Teóricos

Hemos aplicado distintos conceptos teóricos a la hora de crear nuestro videojuego para la NDS. Algunos de ellos, los que hemos considerado más importantes, los detallamos a continuación indicando brevemente a modo de ejemplo el código empleado para ello.

Dispositivos de Entrada y Salida (E/S)

- Pantalla Táctil: la utilizamos tanto como entrada (al tocar encima suyo) y como salida al mostrar imágenes. Para leer si la pantalla táctil es pulsada o no usamos la función `touchRead(&touch)` y establecemos las coordenadas sobre las que dicha función esperará la pulsación o no de la pantalla llamado a los métodos `touch.px` y `touch.py`.

- Timer o Temporizador: el timer se usa para detectar la interrupción que genera el reloj de la consola cada x tiempo y llama a un método (timeout) que va incrementando un contador. Esta es la línea de código usada para llamar al método de incrementación antes mencionado - *irqSet(IRQ_TIMER0, &timeout)*.
- Sonido: Tras haber transformado el archivo de sonido a formato .raw con un máximo de 256 kb, cargamos el archivo mediante *NF_LoadRawSound("sfx/crash", 0, 13025, 1)*. Y simplemente para poder reproducirlo usamos *NF_PlayRawSound(1, 127, 64, true, 0)*. Siendo el número 1 el identificador del sonido para poder implementar varios sonidos. Todo esto tras haber habilitado el sonido con *soundEnable()* y cargado los buffers de sonido con *NF_InitRawSoundBuffers()*.
- PAD y Botones: El PAD lo usamos para controlar el movimiento del sprite. Usamos la función *scanKeys()* para poder detectar cuando una de las flechas de dirección o botón es pulsado, posteriormente la guardamos en una variable *keys = keysHeld()*. Controlamos con un if si la variable *keys* está activada, o lo que es lo mismo, si se ha pulsado alguna tecla del PAD. Si es así y además detectamos que se ha pulsado concretamente en la dirección que queremos (*if (keys & KEY_UP)*) entramos dentro del if para ejecutar la parte del código correspondiente con el fin de que el sprite se mueva en la dirección especificada en la segunda condición del if.

Los botones usados para silenciar la música los controlamos de la misma forma pero controlándolo con otra variable distinta llamada en este caso *newpress*. Controlamos con un if y si se cumplen las condiciones llamamos a las funciones para silenciar la música o para reiniciarla y que vuelva a reproducirse de nuevo, *soundPause(sound_id)* y *soundResume(sound_id)* respectivamente.

- Pantalla Superior: La usamos para mostrar imágenes al igual que la pantalla inferior. Algunos de los comandos que usamos para tratar con las imágenes de fondo son:
 - *NF_InitTiledBgBuffers()* - Inicializa los buffers para almacenar fondos
 - *NF_InitTiledBgSys(0)* - Inicializa los fondos tileados para la pantalla superior.
 - *NF_LoadTiledBg("bg/bg1", "bg1", 256, 256)* - Carga el fondo de la pantalla superior.
 - *NF_CreateTiledBg(0, 3, "bg1")* - Crea los fondos de la pantalla superior.