



Eventos de la botonera y la pantalla táctil en la NDS

Apellidos, nombre	Agustí Melchor, Manuel (magusti@disca.upv.es)
Departamento	Dpto. de Ing. De Sistemas y Computadores
Centro	Escola Tècnica Superior d'Enginyeria Informàtica Universitat Politècnica de València

1 Resumen de las ideas clave

En este artículo vamos a ver cómo se puede interactuar con el usuario en el sentido de cómo implementar la entrada de datos en las aplicaciones para la *Nintendo DS* (NDS) a través de la botonera o de la pantalla táctil.

El interfaz físico de la NDS es amplio, como muestra la Figura 1. La entrada de datos se realiza mayormente con los botones que muestra la Figura 1a. Además de estos, la botonera de la NDS (también denominada *gamepad*), cuenta con una decena de botones como muestra la Figura 1b, junto a un par de “gatillos” en los laterales traseros y uno para la tapa (*lid*).

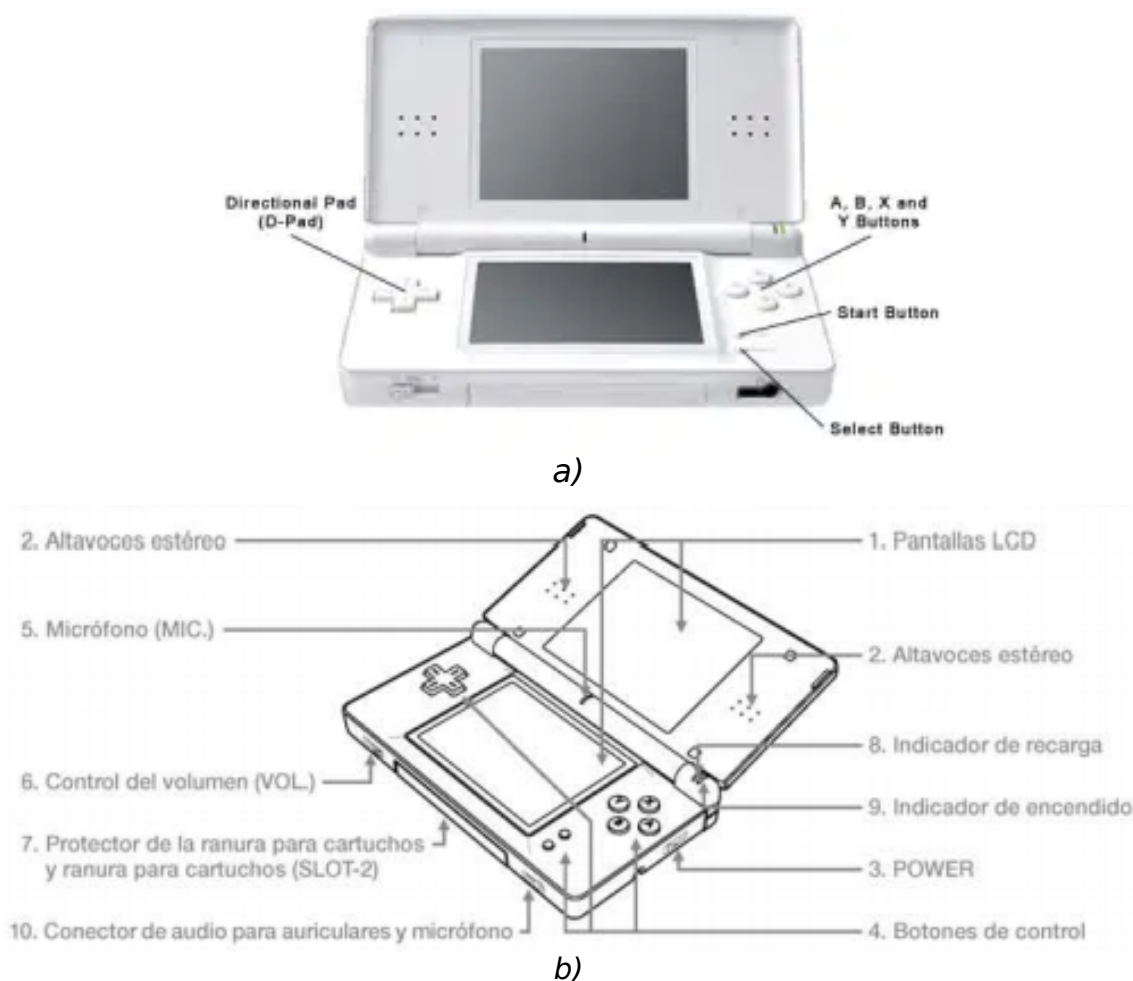


Figura 1: Elementos del interfaz físico de la NDS (a) descripción oficial y (b) detalle de la botonera. Fuente: respectivamente, del “Manual de usuario” <https://www.nintendo.com/consumer/downloads/DSLite_english.pdf> y de <<https://stefanjordan.wordpress.com/2014/05/20/games-controllers/>>.

Además, la existencia de una pantalla táctil ofrece nuevos mecanismos de interacción con el usuario, como: la posibilidad de utilizar un teclado software o el reconocimiento de caracteres manuscritos (habitual en

otros dispositivos portables actuales) y también se puede utilizar el micrófono (en algunos casos se ha utilizado para soplar y en otros para gritar).

En este artículo, nos centraremos en cómo se pueden recoger los eventos de pulsación de la botonera y de la pantalla táctil para interactuar con el usuario. Dejaremos a un lado el uso del micrófono, así como tampoco vamos a considerar ningún dispositivo de expansión de los que se han creado ex profeso (como los que muestra Figura 2).



Figura 2: Ejemplos de aplicaciones de entradas de datos propias: (a) “Easy Piano” <<https://vandal.lespanol.com/juegos/nds/easy-piano/11280#p-29>> y (b) “Guitar Hero” <<https://playconsola.com/juegos-para-el-guitar-hero-ds/>>.

El ejemplo básico que nos inspira apareció en el sitio web de *Drunken Coders*¹ [1]. Estos son un grupo de desarrolladores de la escena *homebrew*² para el desarrollo de aplicaciones para NDS. Esperamos que este trabajo sirva para reconocer el tremendo esfuerzo de estos grandes desarrolladores, máxime dado que en las épocas en que aparecieron no había nada construido. Su trabajo ha servido de inspiración y de base a muchos. ¡Gracias!

Ya no es posible consultar el tutorial original en la red, así que el presente trabajo ofrece una versión de este trabajo actualizada a las versiones existentes de las librerías sobre las que se basa.

Para facilitar el seguimiento de este trabajo se han dispuesto todos los elementos del proyecto que aquí se comenta en *GitHub* [3]. De todas formas, dejaremos los comentarios originales en los ejemplos de código que aquí se mostrarán y una copia en PDF del tutorial original para que quede constancia del trabajo de partida.

¹ Publicado originalmente en la URL <<http://drunkencoders.com/>>, ahora alberga un enlace a un contenido anterior al tutorial en que nos hemos inspirado. En *Github* <<https://github.com/drunken-coders>> ha aparecido un repositorio con la leyenda “Breaking new ground in console homebrew development”. Esperamos que puedan volver a recopilar todos los ejemplos que habían conseguido elaborar en estos años atrás.

² Esto es, que no utiliza en el SDK oficial propuesto por el fabricante de la consola porque tiene tremendas restricciones sobre su difusión y uso.

2 Objetivos

Una vez que el lector haya leído con detenimiento este documento:

- Será capaz de identificar los eventos que genera la botonera y la pantalla táctil de la NDS.
- Será capaz de identificar las instrucciones que permiten el acceso a los eventos de la botonera y de la pantalla táctil de la NDS.
- Será capaz de seguir los ejemplos de código que se proponen como ejercicios de iniciación a la entrada de datos en la NDS.

No es un objetivo instalar las herramientas que permiten la creación del ejecutable así como la carga del mismo en la consola, de hecho, nosotros utilizaremos un emulador (*DeSmuMe* [4]) para generar las capturas. Se recomienda al lector interesado en estas cuestiones, consultar los trabajos de [6] y [7].

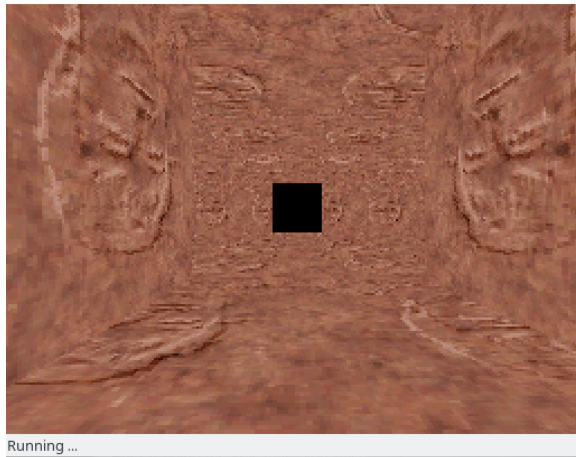
3 Introducción

La NDS tiene unas posiciones en memoria que uno de sus dos procesadores actualiza de forma constante para que el otro pueda procesarlos como eventos que llegan a la aplicación que desarrollamos para esta plataforma.

Con esta funcionalidad podemos dotar de interacción a una aplicación: esto es, podemos dar el control al usuario, son sus acciones las que condicionan la secuencia de nuestra aplicación.

Dependiendo de lo que, además, empleemos para representar la salida tendremos aplicaciones muy diferentes. Así, la Figura 3a muestra desde una aplicación gráfica en la que se mueve la cámara para explorar una escena 3D; la Figura 3b simplifica el interfaz gráfico a 2D con un elemento visual (una esfera roja brillante) que se mueve acorde a la posición del puntero sobre la pantalla táctil; pasando por aplicaciones que hacen uso del teclado software como en la Figura 3c; y hasta aplicaciones de texto que nos muestran (Figura 3d) que podemos obtener, no solo los valores de pulsación en la misma, sino también una valoración de la presión o fuerza que hacemos al presionar la pantalla.

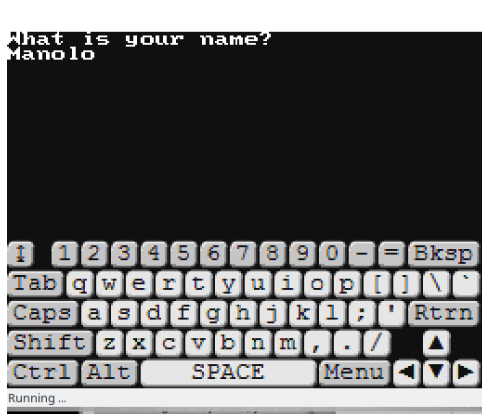
Es difícil hacer programas de demostración interesantes pero simples, así que nos centraremos en aplicaciones de interfaz de texto para explicar cómo leer entrada del usuario. ¿Preparados?



a)



b)



c)



d)

Figura 3: Posibles ejemplos de uso de los controles de la NDS para realizar ejemplos prácticos: desde (a) aplicaciones gráficas 3D a (b) 2D con "sprites", (c) el uso del teclado virtual, hasta (d) interfaces de texto.

4 Ejemplos prácticos de uso

Vamos a ver dos ejemplos. El primero nos mostrará cómo obtener el estado de la botonera de la NDS: se realiza en base a comprobar el estado de cada una de las teclas. Este estado nos dirá si está siendo presionada o si se ha soltado. Y pasaremos a identificar la tecla o teclas asociadas a cada estado para proceder según lo que corresponda en nuestra aplicación.

El segundo caso nos mostrará cómo obtener la información relativa a la pantalla táctil, lo que utilizaremos para pintar en la pantalla. Veremos cómo averiguar si se ha tocado la pantalla y en qué posición se ha hecho.

Toda esta información la mantiene el *hardware* de la NDS, el procesador ARM7 es el encargado de actualizar los valores de tecla pulsada, soltada o mantenida entre cada ciclo de refresco de pantalla. Esa es la máxima

precisión o resolución temporal, que puede tener nuestra gestión de la entrada de datos. Además, también se registra la pulsación de un punto en la pantalla táctil (con el *stylus* o con el dedo) y la presión o fuerza de este gesto.

4.1 La interacción básica: uso de la botonera

Nuestra primera demostración, véase el Listado 1, verifica las pulsaciones de teclas e imprime su estado en la pantalla, por eso (en el tutorial original), este ejemplo se llama “Talking to the keyboard”. Para ello, en la línea 5 se inicializa una de las pantallas para funcionar en modo consola de texto y, así, que las instrucciones *printf* puedan escribir en la pantalla. Podemos ver una captura de las pantallas de la consola que muestran esta situación en la Figura 4a.

```
1  #include <nds.h>
2  #include <stdio.h>
3
4  int main(void) {
5      consoleDemolnit();
6      while(1) {
7          scanKeys();
8          int held = keysHeld();
9          if( held & KEY_A) printf("Key A is pressed\n");
10         else printf("Key A is released\n");
11         if( held & KEY_X) printf("Key X is pressed\n");
12         else printf("Key X is released\n");
13         if( held & KEY_TOUCH) printf("Touch pad is touched\n");
14         else printf("Touch pad is not touched\n");
15
16         swiWaitForVBlank();
17         consoleClear();
18     }
19     return 0;
20 }
```

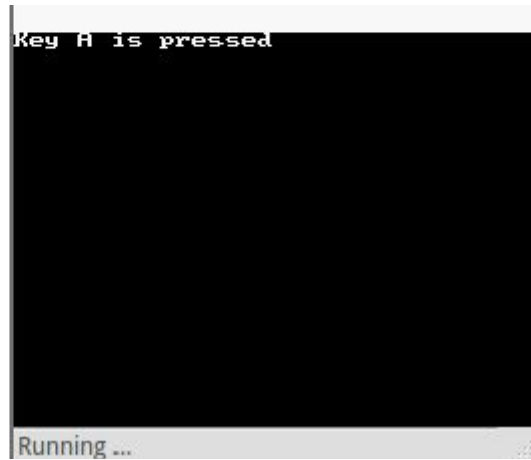
Listado 1: “Talking to the keyboard”, gestión de los eventos básicos de la botonera y la pantalla táctil [1].

Como podemos observar, en cada iteración del bucle (línea 6) se lee el estado de la botonera con la instrucción *scanKeys* (línea 7) y se obtienen las que están mantenidas pulsadas con *keysHeld* (línea 8). El valor que devuelve esta instrucción corresponde con una palabra de 32 bits en la que cada tecla tiene un bit asignado y, ese bit, está a ‘1’ si la tecla correspondiente había sido pulsada y se mantiene así desde la última llamada a *scanKeys* o ‘0’ en caso contrario.

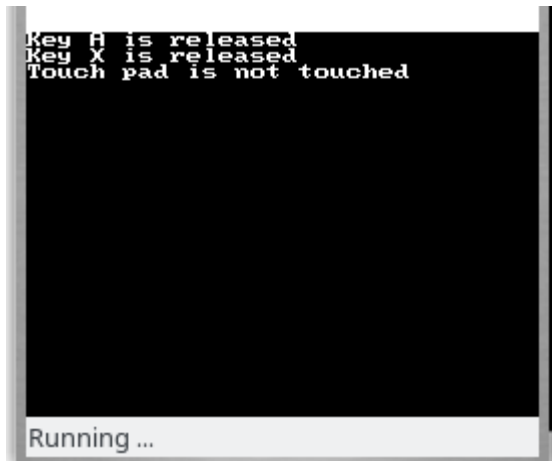
Para consultar cada bit, el bucle principal verifica la pulsación de las teclas (o botones) ‘A’ y ‘X’, línea 9 a la 12. Esa misma secuencia es todo lo que hay que hacer para verificar si se presiona cualquiera de los botones de la NDS, cambiando en su caso el identificador de la tecla que queremos comprobar. En la Figura 4b y siguientes se muestran los mensajes de tecla recientemente pulsada o soltada.



a)



b)



c)



d)

Figura 4: Capturas de diferentes momentos de la ejecución del ejemplo del Listado 1: (a) Inicio, (b) detección de eventos de la botonera y (c y d) detección de eventos de botonera y pantalla táctil.

Y, con la misma estrategia, las líneas 13 y 14, nos informan de si se ha tocado en algún punto la pantalla. Hemos de hacer algo más para averiguar dónde, lo veremos en el ejemplo del apartado 4.2 La pantalla táctil permite “tocar cosas”. De momento, podemos ver en la Figura 4c y Figura 4d que podemos detectar los mismos eventos que con las teclas, en este caso referidos a si la pantalla acaba de registrar una pulsación o si se acaba de levantar el dedo (o el lápiz) con el que se tocaba la pantalla

Las líneas 16 y 17 esperan a que la pantalla termine de dibujarse y luego se borra la pantalla. Esto evita parpadeos en la aparición de contenido en la pantalla.

También podemos registrar otros eventos relacionados con los ya mencionados, se puede utilizar: `keysDown`, que solo tendrá los bits a ‘1’

de las teclas que se presionaron desde la última llamada a `scanKeys`; y también existe un `keysUp` para las teclas liberadas.

4.2 La pantalla táctil permite “tocar cosas”

La pantalla táctil es una característica increíble para un sistema de videojuegos portable en la época en que apareció la NDS, ahora lo podemos encontrar en multitud de dispositivos y hasta parece que lo echamos de menos en la plataforma de escritorio. La opción táctil no solo permite una experiencia de juego interesante, sino que también agrega un nuevo nivel de diversión al juego al ofrecer un interfaz más natural y directo.

En esta sección presentaremos cómo funciona la pantalla táctil y mostraremos un ejemplo de uso de los eventos que se generan y la lectura de las posiciones de pantalla.

La pantalla táctil de la NDS utiliza [9] un recubrimiento resistivo que cambia la conducción según el área del objeto de contacto. Este cambio se mide mediante algunas conversiones analógicas a digitales en un chip especial dentro de la NDS y traducido a una ubicación X e Y. Estas medidas también pueden usarse para determinar el área del punto de contacto que, hasta cierto punto, puede traducirse por un valor de “presión” ejercida sobre la pantalla y así, p. ej., simular el pintar con un pincel cuya “huella” depende de la fuerza con que aplica sobre la superficie en la que se pinta.

```
1  #include<nds.h>
2  #include<stdlib.h>
3  int main(void) {
4      touchPosition touch;
5
6      videoSetMode(MODE_FB0);
7      vramSetBankA(VRAM_A_LCD);
8      //notice we make sure the main graphics engine renders
9      //to the lower lcd screen as it would be hard to draw if the
10     //pixels did not show up directly beneath the pen
11     lcdMainOnBottom();
12     while(1) {
13         scanKeys();
14         if(keysHeld() & KEY_TOUCH) {
15             // write the touchscreen coordinates in the touch variable
16             touchRead( &touch );
17             VRAM_A[touch.px + touch.py * 256] = rand();
18         }
19     }
20     return 0;
21 }
```

Listado 2: “Touching things”, características avanzadas del uso de la pantalla táctil [1].

Ahora, para una demostración simple, le mostraremos el programa de arte más simple posible: pintará puntos de colores (escogidos estos al azar) donde sea que toquemos la pantalla.

Esta segunda demostración, véase el Listado 2, comprueba que se ha tocado la pantalla táctil y averigua las coordenadas donde se ha producido el evento, por eso (en el tutorial original), este ejemplo se llama "Touching things".

Los valores de las coordenadas donde se ha tocado la pantalla se leen usando la función *touchRead* (línea 16), que devuelve las coordenadas en analógico (*raw*) donde se ha producido el contacto y las coordenadas transformadas en valores de píxel.

Y, como se ha comentado en el apartado 4.1 La interacción básica: uso de la botonera, también puede obtener un evento cuando se está usando el lápiz (stylus) o el dedo, para tocar la pantalla o al levantarlo. Esto le indica si el lápiz está arriba o abajo para que sepa cuándo leer el panel táctil. El uso de *keysHeld* (línea 14) servirá para detectar cuando el lápiz o el dedo, toca la pantalla por primera vez y mientras que no se levante. Esto nos permite dibujar nuestros puntos en la pantalla mientras no se levante el bolígrafo. Así hemos dibujado el contenido de la Figura 5. Que hayamos dibujado letras, u otras cosas, no es significativo para la aplicación. Dese cuenta el lector que podríamos llegar a realizar un reconocedor de caracteres manuscritos, tomando el último trazo realizado y comparando con una lista de posibles dibujos de cada letra. Pero esto es otra historia.

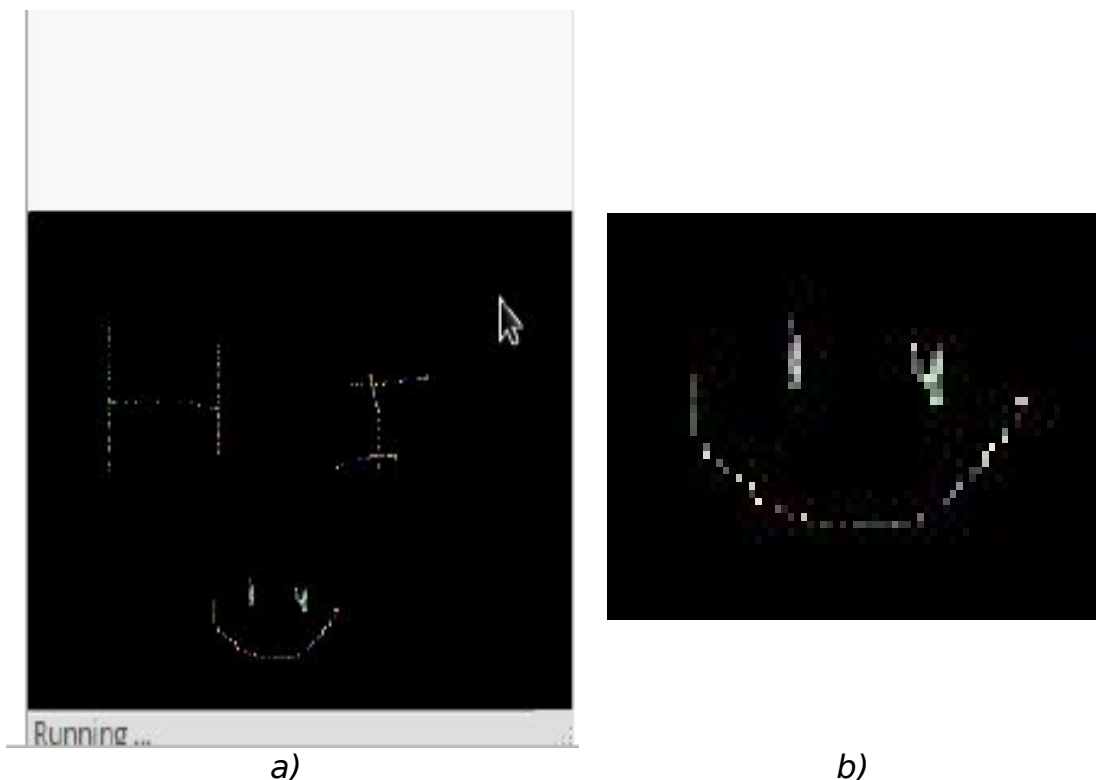


Figura 5: Ejemplo de salida del Listado 2: (a) la pantalla completa y (b) un detalle para ver que cambian los colores.

Volviendo a nuestro ejemplo. ¿cómo se dibujan los puntos? En primer lugar, la inicialización de la consola (línea 6) establece un modo gráfico para el motor principal, que se asigna a la pantalla inferior de la NDS (línea 11). Además, se ha configurado (línea 7) esta pantalla con un área

de memoria (VRAM_A), por lo que podremos acceder a cada punto de la pantalla en una posición de esa área de la memoria de vídeo (VRAM). Construiremos la imagen a base de modificar, punto a punto, el contenido de esa zona de memoria, utilizando el lápiz (o el dedo) a modo de pincel.

En segundo lugar, al usar la función *touchRead* (línea 16), la variable *touch* contendrá la posición de columna (*px*) y fila (*py*) en que se produce la pulsación en pantalla. La resolución de las pantallas de la NDS es de 256x192 (columnas x filas) y dado que la memoria de vídeo está organizada en memoria por filas, la expresión de la línea 17 asigna un color al punto de la memoria de vídeo que corresponde con la posición de la pulsación.

El color se selecciona al azar usando la función *rand*. Podríamos modificar el código para hacer variar el color con la pulsación de una tecla y tener así una forma de elegir el color.

Al dibujar rápidamente, probablemente notaremos grandes espacios entre los puntos. Esto se debe a que las coordenadas táctiles solo se actualizan una vez por cuadro y podemos haber movido el *stylus* (o el dedo) mucho más rápido que eso. ¡Cuida el trazo, no corras!

5 Conclusiones y cierre

En este artículo hemos visto cómo se puede interactuar con el usuario en el sentido de cómo implementar la entrada de datos en las aplicaciones para la NDS. Hemos revisado las características del interfaz físico para centrar nuestra atención en la botonera y la pantalla táctil. Y, sobre estos elementos, hemos visto cómo se pueden recoger los eventos de pulsación de la botonera y de la pantalla táctil para interactuar con el usuario.

Para ofrecer una parte práctica nos hemos basado en los ejemplos que se publicaron originalmente en el sitio web de *Drunken Coders* [1] y que creemos importante rescatar, puesto que detrás de este grupo está un buen número de las personas que han hecho posible el desarrollo de la escena *homebrew* para NDS. Para facilitar el seguimiento de este trabajo y mantener la disponibilidad de estos ejemplos, se han dispuesto todos los elementos que aquí se comentan en *GitHub* [3].

Quiero ahora animar al lector a hacer suyos los ejemplos e introducir modificaciones en ellos: p. ej. el segundo se presta muy rápidamente a añadir la opción de cambio de color o tamaño del trazo para dibujar. ¿Lo intentamos?

6 Bibliografía

[1] Dovoto. (2015). Day 2 - A Hardware Overview

[2] J. D. Jaén Gomariz. (2015). Tutorial práctico para desarrollo de videojuegos sobre plataforma Nintendo NDS. Disponible en <http://hdl.handle.net/10251/56433>.

- [3] M. Agustí, (2020). Repositorio del proyecto “NDS-homebrew-development”. Disponible en <<https://github.com/magusti/NDS-hombrew-development>>.
- [4] DeSmuME. Página web del proyecto. Disponible en <<http://desmume.org/>>.
- [5] *Grit. GBA Raster Image Transmogriker*. Disponible en <<http://www.coranac.com/projects/grit> >.
- [6] J. Amero (Patater). (2008). Introduction to Nintendo DS Programming. Disponible en <<https://patater.com/files/projects/manual/manual.html>>.
- [7] .GBATEK. Gameboy Advance / Nintendo DS - Technical Info. Disponible en <<https://www.akkit.org/info/gbatek.htm>>.
- [8] F. Moya y M. J. Santofimia. (2011). Laboratorio de Estructura de Computadores empleando videoconsolas Nintendo DS. Ed. Bubok Publishing. ISBN. 978-84-9981-039-3.