

Iniciación a la robótica con LEGO® MINDSTORMS, 12ª entrega

Uso social de robots LEGO® MINDSTORMS

Por Koldo Olaskoaga

En julio tuve la oportunidad de participar en el curso "Caring robots" organizado por las universidades Windesheim Flevoland de Almere (Holanda) y La Salle - Universitat Ramon Llull de Barcelona. En dicho curso, entre otras cosas, se habló de lo que pueden aportar las actividades con robots en la rehabilitación de niños con daño cerebral y en la mejora de las habilidades sociales de los niños autistas. La Salle - Universitat Ramon Llull está colaborando con Tufts University, Massachusetts (USA), Deusto y Comillas en una investigación que trata de determinar si el uso de robots en la rehabilitación de niños con daño cerebral ofrece una ventaja respecto a los métodos convencionales.

Una de las tareas propuestas en el curso fue crear una actividad que posteriormente sería testada con niños que padecen daño cerebral en el hospital Sant Joan de Déu de Barcelona, principal colaborador en los proyectos anteriormente comentados de La Salle.

A continuación se presenta la actividad, centrando las explicaciones principalmente en el aspecto de la programación.

La actividad

Una de las ideas propuestas era hacer un juego que ayude a ejercitar la memorización. Tras reflexionar sobre ello y analizar las posibilidades que ofrece LEGO® MINDSTORMS Educación, que era el material del que disponíamos, se decidió crear un juego que requiriese recordar secuencias de sonidos. El robot debía de reproducir secuencias de hasta 4 notas (una por puerto de entrada en el NXT) y el niño debía de repetirlas a continuación vía el teclado creado al efecto. Se consideró que el recordar secuencias de notas, por el hecho de ser sonidos, podía ser complicado para algunos niños, así que se decidió combinarlos con otro tipo de estímulo, en este caso con colores. En la caja de LEGO MINDSTORMS Educación, se pueden encontrar tres lámparas que se pueden montar con piezas transparentes de color, lo que permitió incorporar tres lámparas (roja, verde y amarilla) que se encendían al reproducir las notas. Tener tres lámparas y tres puertos de salida en el NXT nos llevó a reducir el teclado a tres teclas.

¿Qué pasaba si acertaban la secuencia? Además de los aplausos y cara sonriente en el display del NXT se pensó que sería interesante incorporar algún tipo de recompensa que agregase diversión. Así que se montó un robot móvil con un sensor de luz que avanzaba o retrocedía sobre un fondo blanco con líneas negras transversales, si acertaba avanzaba, y si fallaba retrocedía. El objetivo del juego era llegar hasta la última línea.

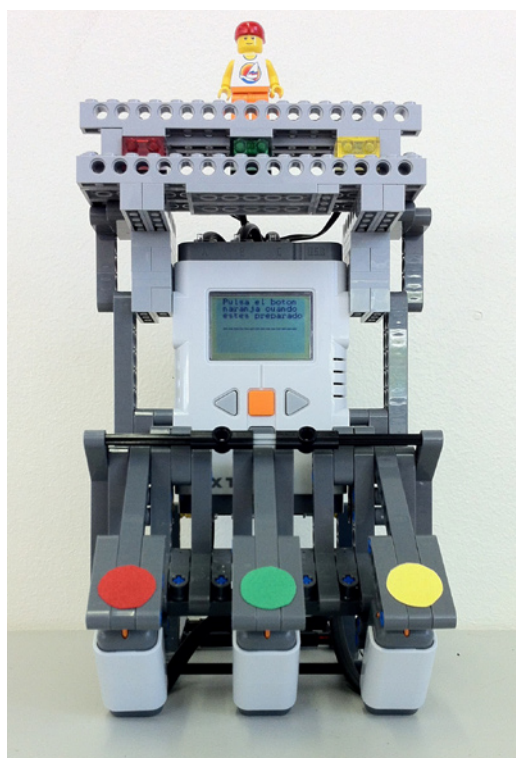
La consola de juegos

A la hora de montar la consola de juegos, los requerimientos fueron los siguientes:

- Tres sensores de contacto con teclas con suficiente tamaño

- y separación para ser utilizadas por niños.
- Tres lámparas con filtro de color alineadas con las teclas.
- Una estructura que permita mantener la consola erguida sobre una mesa con un fácil acceso a la pantalla y teclas del NXT.

Al no disponer de piecero de los tres colores, se puso una pegatina de color en cada tecla para facilitar su identificación.



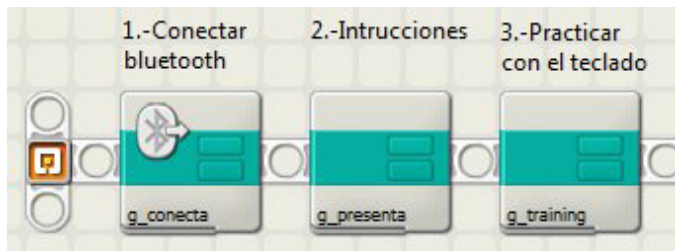
El programa

No voy a entrar aquí a fondo en el programa, ya que alargaría en exceso el artículo, pero sí voy a comentar algunos de los aspectos que considero pueden ser de interés.

El programa de la consola tenía las siguientes partes:

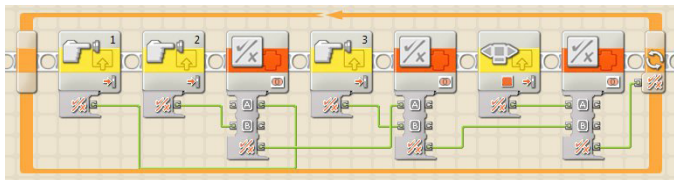
1. Establecer la conexión vía bluetooth
2. Presentar las instrucciones de uso
3. Practicar con el teclado: su función es conocer y practicar con el teclado. Al pulsar una tecla reproduce un sonido y enciende la lámpara del color correspondiente. Al pulsar el botón naranja el programa sigue adelante.
4. El juego
 - a. Selección de nivel: Entrenamiento (secuencia de 3 notas/colores), Básico (4), Medio (5) y Avanzado (6[1])
 - b. El juego propiamente dicho: genera secuencia aleatoria; la reproduce; registra las pulsaciones; comprueba el resultado y da el feedback

El programa se creó de modo modular, de tal forma que era posible testear cada parte antes de pasar a la siguiente. Se crearon nuevos bloques en todos los casos en que fue posible. En la siguiente imagen pueden verse los tres primeros pasos del programa, cada uno convertido en un nuevo bloque.

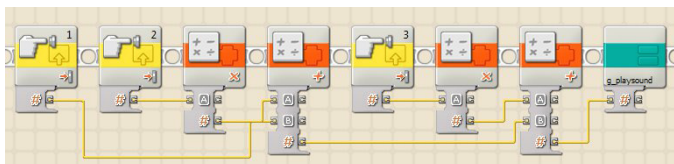


Vamos a dejar de lado los dos primeros pasos y centrarnos en el tercero. En este paso se ofrece practicar con el teclado y comprobar su funcionamiento el tiempo que haga falta. El programa continúa y da inicio al juego al pulsar el botón naranja del NXT.

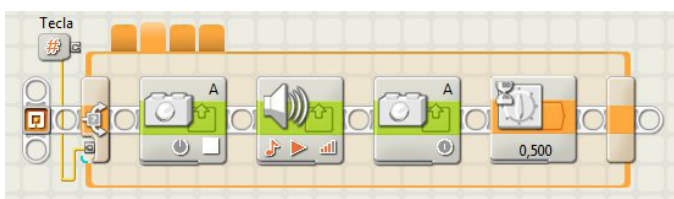
Para ello, el NXT ha de leer los tres sensores de contacto y el botón naranja de modo secuencial hasta que uno de ellos sea pulsado. Utiliza la operación booleana OR que determina si alguno de los sensores ha sido pulsado. Si ha sido así, sale del bucle y reproduce el sonido o continúa con el programa.



Una vez que se ha pulsado una de las teclas, o el botón naranja del NXT, se reproducirá la nota correspondiente, se encenderá una lámpara o finalizará la parte de prácticas. Este paso se podría hacer con un Condicional (Si.. Sino...), pero para evitar anidar uno dentro de otro convertimos la pulsación en un número: 1, 2 o 3 dependiendo de la tecla pulsada. Esto se puede hacer por medio del siguiente fragmento de programa:



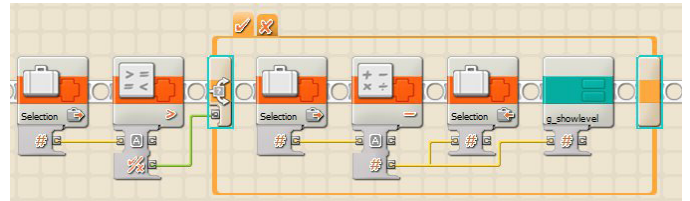
Aprovechamos una de las salidas del bloque Leer Sensor de Contacto que da un valor 1 si hay contacto y 0 si no lo hay. Dejando el valor del primero en 0 ó 1, multiplicando el segundo por 2 y el tercero por 3 y haciendo la suma de estos valores sabremos qué tecla se ha pulsado. En caso que haya sido pulsado el botón Naranja, el resultado será 0. Al final se puede ver el bloque que reproduce el sonido y enciende la lámpara, dado que se va a utilizar en diferentes partes del programa se ha convertido en un nuevo bloque. Su contenido es el siguiente:



Una vez que se ha practicado, llega el juego. Lo primero que hay que hacer es seleccionar el nivel en el que se desea jugar.

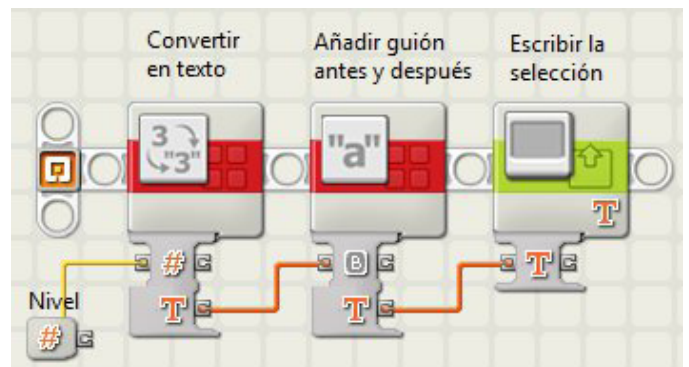
Para ello se utilizan las teclas grises para seleccionar el nivel y la naranja para confirmar la elección y comenzar el juego. Los pasos a seguir serán los siguientes:

1. Mostrar las opciones
2. Crear y dar un valor inicial a una variable que almacenará el nivel elegido.
3. Mostrar el nivel correspondiente a dicho valor
4. Iniciar un bucle que no finalizará hasta que se pulse el botón naranja:
 - a. Esperar a que se pulse uno de los tres botones (naranja o flechas grises)
 - b. Si es una de las flechas sumar o restar una unidad a la variable y representar la nueva selección (tener en cuenta que hay cuatro niveles y el valor deberá estar comprendido entre 1 y 4. Hay que tener cuidado de que al restar o sumar no nos salgamos del rango y el resultado no sea ni 0 ni 5.



En la imagen puede verse el código que se ejecuta al pulsar la flecha izquierda. Si el contenido de la variable Selection es mayor que 1 se le resta una unidad y se muestra en pantalla, en caso contrario se deja tal y como estaba.

A la hora de representar las opciones, y la selección actual, hay que tomar ciertas precauciones. El bloque Visualizar tiene una casilla de verificación que si está seleccionada, borra todo lo anterior. Para visualizar el nivel podría utilizarse un condicional, pero se puede hacer de un modo más directo utilizando los bloques que permiten operar con texto. Cuando se cambia la opción, el valor numérico se convierte en texto, para a continuación añadir por delante y por detrás un guión (es una cuestión estética). La cadena de texto resultante se escribe en la línea 8 sobre la elección anterior.

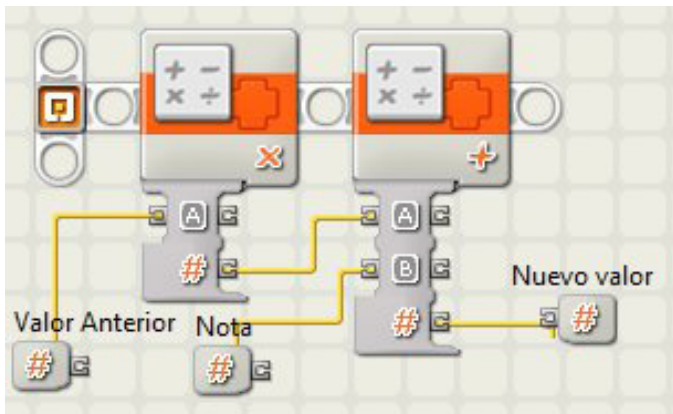


Una vez que se ha seleccionado el nivel toca jugar, para lo que primero elige de modo aleatorio una de las tres notas, la reproduce (enciende a la vez la lámpara correspondiente) y la almacena. Así tantas como las que correspondan al nivel seleccionado.

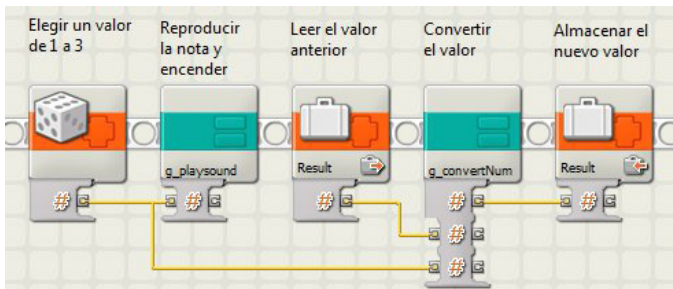
El modo más directo de almacenar una secuencia de notas sería una matriz, pero dado que NXT-G no las soporta de modo estándar optamos por otro modo. Lo hicimos utilizando una única variable numérica en la que la nota (un valor de 1 a 3) se almacena en la posición de unidades, decenas, centenas... Para ello, una vez generada la nota por medio del bloque Aleatorio (con un valor de 1 a 3), se realiza el siguiente cálculo para almacenarla:

$$\text{Nuevo valor} = \text{Nota} * 10 + \text{Valor anterior}$$

[1] g_GenerateSo: Este bloque genera la secuencia de sonidos. La entrada es el nivel de 1 a 4 y la salida es un número que representa la secuencia de entre 3 y 6 sonidos/colores.



Así que para cada nota se ejecutarán los pasos que se ven en la figura:

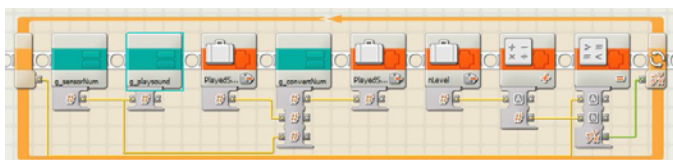


Aquí se reutiliza el bloque `g_playsound`, ya visto anteriormente, y se crea uno nuevo para convertir el valor, tal y como acabamos de ver.

Cuando el usuario introduce una secuencia vía teclado se realiza la misma operación, de tal modo que al final se puede comparar con el valor que representa la secuencia a reproducir.

El bucle que permite al usuario introducir la secuencia consta de los siguientes pasos:

1. Esperar a que se pulse una tecla y convertir el valor en un número de 1 a 3 (bloque `g_sensorNum`).
2. Reproducir el sonido y encender la lámpara correspondiente (bloque `g_playsound`).
3. Registrar la elección del modo que se ha visto anteriormente (bloque `g_convertNum`), en este caso la variable es `PlayedSound`.
4. Sumar una unidad a la variable `nLevel` (es la que almacena el nivel con un valor de 1 a 4) y compararla con el número de bucles completados. Si el resultado es Verdadero quiere decir que ha finalizado la secuencia.



Este bucle se tiene que ejecutar más o menos veces dependiendo del nivel: el valor que representa el nivel + 2 veces, es decir, entre 3 y 6 veces. Para llevar la cuenta, en lugar de crear una variable que actúe como contador, utilizamos el propio contador del bucle. La toma del contador se muestra a la izquierda del bucle (es necesario seleccionar la casilla de verificación correspondiente en el panel de configuración del bloque Bucle para que lo muestre). El valor que se obtiene en esta toma es el número de bucles completados. Esa es la razón de que se compare el valor del contador con el valor de `nLevel + 1`.

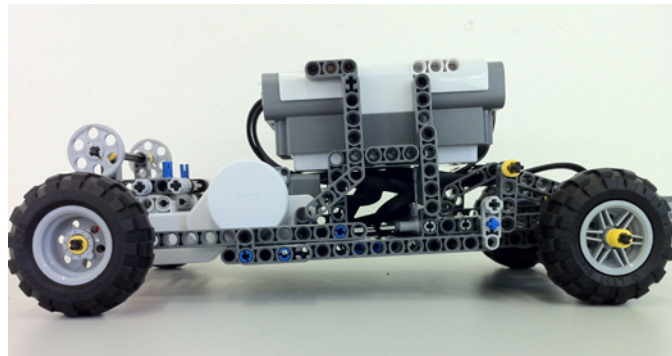
Ahora ya solo queda comparar la secuencia reproducida con la generada previamente. Si ha acertado, se muestra en la

pantalla una cara sonriente y se escuchan aplausos, en caso contrario, cara triste y un sonido que sugiere error. Además envía un verdadero o falso al vehículo vía bluetooth.

El programa original daba 5 oportunidades para conseguir con el vehículo llegar a la meta.

El vehículo

El vehículo fue un robot móvil muy sencillo, ya que lo único que se le pedía era la capacidad de avanzar y retroceder hasta la siguiente línea. Así que, incluso, se podía haber montado con un solo motor más un sensor mirando hacia abajo.



El programa está a la espera de recibir un verdadero o falso de la consola. Una vez recibido, si el valor del mensaje es Verdadero avanza hasta la siguiente línea, mientras que si es Falso retrocede (a no ser que esté en la línea de salida). El programa registra el número de intentos y la posición, de tal modo que cuando alcanza los 5 intentos retrocede a la salida, tanto haya llegado a la meta como no (en el primer caso tras reproducir un sonido de júbilo).

Es importante que el robot controle en qué parte de la línea se encuentra, es decir, si para ir a la siguiente ha de atravesar primero la línea o no. Fue algo que dio muchos problemas a los que programaron el vehículo.

Puesta en práctica

La actividad se testeó en el Hospital de Sant Joan de Déu de Barcelona con varios niños con daño cerebral. Si bien las condiciones del test no fueron las necesarias para sacar conclusiones, se observó que los niños pequeños se centraban totalmente en el vehículo olvidándose de la otra parte del juego. Por otra parte, en algunos casos tres notas/colores resultaban excesivos para empezar, con dos teclas y secuencias de dos notas/colores hubiese sido suficiente. Desde el punto de vista humano fue una experiencia muy enriquecedora, gracias a la colaboración del hospital y las familias implicadas.

Posible mejoras

- Uso del teclado para el control de menús para manos menos hábiles.
- Si bien fue algo que no se planteó siquiera, el robot móvil al llegar a la meta podría coger algo y traérselo al usuario como recompensa.

Esta actividad fue el resultado de un trabajo en equipo en el que participamos Chang Long Zhu, Juan Pablo Forero, Louellen Palm y yo mismo.

#



Lrobotikas.net

Robótica Educativa y Recreativa