

ADT_joystick API

Conexión con el puerto de juegos en C++

Mario Chirinos Colunga
Áurea - Desarrollo Tecnológico

18 de junio de 2013

Índice

1. Introducción	1
2. ADT_joystick API	1
2.1. Requisitos	1
2.2. Listado de dispositivos	2
2.3. Conexión y desconexión	2
2.4. Información del dispositivo	2
2.5. Eventos	3
3. Ejemplo	3
3.1. Requisitos	3
4. Notas	3

1. Introducción

En este documento se describe nuestra interfaz de programación de aplicaciones para la conexión con *joystick* o *gamepads* conectados a un puerto de juegos. Se utiliza *glib* para monitorizar el estado del puerto de juegos y generar eventos. El documento está estructurado de la siguiente manera: en la sección §2 describe el funcionamiento de nuestro código, la sección §3 describe detalles relacionados al ejemplo incluido con el código fuente y la sección §4 proporciona notas y comentarios finales sobre este documento.

2. ADT_joystick API

2.1. Requisitos

Para compilar código de esta API se requiere contar con el paquete **libgtk2.0-dev**, agregar al compilador C++ las banderas obtenidas mediante: **pkg-config --cflags gtk+-2.0** y contar con un núcleo de Linux versión 2.4 o superior.

```

#ifndef ADT_JOYSTICK_H
#define ADT_JOYSTICK_H
//-----
#include <glib.h>
#include <linux/joystick.h>
#include <vector>
using namespace std;
typedef void (*jsPt2Function)(unsigned int number, int value, void *userdata);
//-----
class JsData
{
public:
    char name[80];
    int nAxis;
    int nButtons;
    JsData();
};
//-----
class ADT_joystick
{
public:
    int enumJoysticks();
    int connect(const char *filename);
    int readJs();
    void setAxisCallback(jsPt2Function _fpt);
    void unsetAxisCallback();
    void setButtonCallback(jsPt2Function _fpt);
    void unsetButtonCallback();

    vector<JsData> getList() const;
    const char* getDevName() const;
    const char* getName() const;
    unsigned int getAxisCount() const;
    unsigned int getButtonCount() const;
    int getAxis(unsigned int i) const;
    bool getButton(unsigned int i) const;

    ADT_joystick();
    ~ADT_joystick();
};
//-----
#endif

```

Figura 1: ADT_joystick.h

2.2. Listado de dispositivos

Para en listar los puertos de juegos conectados al sistema se utiliza la función miembro `int enumJoysticks(void)` la cual devuelve el numero de dispositivos encontrados. La lista creada es del tipo `vector<JsData>`. La función `getList()` permita acceder a esta lista.

2.3. Conexión y desconexión

La función miembro `int connect(const char *filename)` permite conectarse el puerto de juegos especificado por *filename*, e.g. `connect(/dev/input/js0)`. La función devuelve cero si la conexión fue exitosa.

2.4. Información del dispositivo

Las siguientes funciones permiten obtener información acerca del puerto de juegos al que se esta conectado.

- `getDevName()` Regresa el nombre del puerto de juegos, e.g. */dev/input/js0*.
- `getName()` Regresa el nombre de identificación del puerto de juegos asignado por el fabricante.
- `getAxisCount()` Regresa el numero de ejes del puerto de juegos.
- `getButtonCount()` Regresa el numero de botones del puerto de juegos.

- `getAxis(unsigned int i)` Regresa el valor del eje *i*.
- `getButton(unsigned int i)` Regresa el valor del botón *i*.

2.5. Eventos

La API utiliza [glib](#) para monitorizar el estado del puerto de juegos y llamar a una función de retro llamada (*callback*) cuando se genere un evento. Se debe iniciar un *gtk main loop* en el programa principal par activar los eventos.

Las funciones miembro `setAxisCallback(fpt)` y `setButtonCallback(fpt)` permiten conectar una retro llamada a un evento generado por un botón o un eje del puerto de juegos. La función a conectar debe de tener la firma `fpt(unsigned int number, int value, void *userdata)`. Las funciones miembro `unsetAxisCallback()` y `unsetButtonCallback()` permiten desconectar las retro llamadas.

3. Ejemplo

Junto con el código de esta API se provee de un programa ejemplo para demostrar su funcionamiento.

3.1. Requisitos

Para compilar el programa ejemplo se requiere contar con el paquete **libgtk2.0-dev**. Las banderas de compilación se especifican en el archivo *makefile* de este ejemplo.

4. Notas

El código fuente de esta API puede ser descargado en [nuestro sitio web](#), en donde también se pueden reportar errores en el código fuente. Para reportar errores en este documento favor de escribir a errata@aurea-dt.com.

Para futuras versiones de esta API se contempla la calibración y la retroalimentación háptica del joystick.