

ADT_rovioAPI

Interfaz de aplicación de código abierto en C++ para el control de Rovio™ - WowWee®

Mario Chirinos Colunga
Área - Desarrollo Tecnológico

26 de junio de 2013

Índice

1. Introducción	1
2. ADT_rovioAPI	1
2.1. Requisitos	2
2.2. Control de acciones	2
2.3. rovioStreaming	2
2.3.1. Conexión y desconexión	2
2.3.2. Manipulación de los datos del vídeo	3
2.3.3. Manipulación de los datos del audio	3
2.4. rovioSpeak	3
3. Ejemplo	5
3.1. Requisitos	5
4. Notas	5

1. Introducción

Este documento está estructurado de la siguiente manera: en la sección §2 describe el funcionamiento de nuestra 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_rovioAPI

La clase principal de la API es `ADT_rovioAPI` esta se deriva dos clases base: `rovioStreaming` para acceder a la cámara IP utilizando `GStreamer` y `rovioSpeak` para síntesis de voz utilizando `eSpeak` y `curl`. `ADT_rovioAPI` también provee de funciones para enviar comandos a rovio utilizando `curl`.

2.1. Requisitos

Para compilar código de esta API se requiere contar con los siguientes paquetes:

- libgnomeui-dev
- libgstreamer0.10-dev
- libgstreamer-plugins-base0.10-dev
- libcurl4-gnutls-dev
- libespeak-dev

2.2. Control de acciones

La API provee de las siguientes funciones para controlar las acciones de rovio:

- `setHome()` Define la posición actual de rovio como la enfrente de la base de carga.
- `goHome()` Ordena a rovio a ir a la base de carga.
- `stop()` Ordena a rovio detenerse.
- `move(angle, speed)` Ordena a rovio moverse en la dirección señalada por *angle* y a la velocidad indicada por *speed*.
- `rotate(angle, speed)` Ordena a rovio rotar *angle* grados a velocidad *speed*.
- `headUp()` Cabeza arriba.
- `headMiddle()` Cabeza enmedio.
- `headDown()` Cabeza abajo.
- `light(1)` Encender/apagar luz.

2.3. rovioStreaming

Esta clase utiliza GStreamer para acceder a la transmisión de audio y vídeo de la cámara IP de rovio. La figura 1 muestra la estructura de la línea de flujo de los datos de audio y vídeo a través de los elementos de GStreamer.

2.3.1. Conexión y desconexión

La función miembro `connect()` permite conectarse a la cámara IP de rovio. Para desconectarse utiliza la función `void disconnect()`.

2.3.2. Manipulación de los datos del vídeo

La API permite acceder y modificar el bloque de memoria en donde se encuentra la imagen obtenida por el la cámara IP de rovio. Para poder manipular este bloque de memoria primero se debe de crear una función con la firma `void filtro(unsigned char *data, unsigned int width, unsigned int height, unsigned int nChannels, void* userdata)`, en donde el puntero *data* apunta al bloque de memoria, *width*, *height*, *nChannels* son las dimensiones de la imagen y *userdata* apunta a **this**. Esta función será ejecutada cada vez que el dispositivo de vídeo adquiere una nueva imagen, en ella se puede manipular el bloque de memoria. Las funciones que permiten controlar el acceso a la nueva función son:

- `int setVideoFilter(pt2Function)` modifica el puntero *videoFpt* con la dirección de la función en el parámetro de entrada.
- `int unsetVideoFilter()` *videoFpt* = NULL.
- `int startVideoFilter()` Conecta la señal "*signal-handoffs*" del dispositivo de vídeo a la función especificada por `setFilter`. Esta señal es emitida cuando el dispositivo de vídeo entrega el buffer de memoria.
- `int stopVideoFilter()` Desconecta la señal "*signal-handoffs*".

2.3.3. Manipulación de los datos del audio

La API permite acceder y modificar el bloque de memoria en donde se encuentra el audio obtenido por el la cámara IP de rovio. Para poder manipular este bloque de memoria primero se debe de crear una función con la firma `void filtro(unsigned char *data, unsigned int width, unsigned int height, unsigned int nChannels, void* userdata)`, en donde el puntero *data* apunta al bloque de memoria, *width*, *height*, *nChannels* son las dimensiones de la imagen y *userdata* apunta a **this**. Esta función será ejecutada cada vez que el dispositivo de vídeo adquiere una nueva imagen, en ella se puede manipular el bloque de memoria. Las funciones que permiten controlar el acceso a la nueva función son:

- `int setVideoFilter(videoCallback)` modifica el puntero *videoFpt* con la dirección de la función en el parámetro de entrada.
- `int unsetVideoFilter()` *videoFpt* = NULL.
- `int startVideoFilter()` Conecta la señal "*signal-handoffs*" del dispositivo de vídeo a la función especificada por `setFilter`. Esta señal es emitida cuando el dispositivo de vídeo entrega el buffer de memoria.
- `int stopVideoFilter()` Desconecta la señal "*signal-handoffs*".

2.4. rovioSpeak

Esta clase utiliza `espeak` para sintetizar audio y enviarlo a Rovio utilizando `libculr`. La función miembro `void speak(const char* text)` sintetiza el texto *text* y lo envía a Rovio.

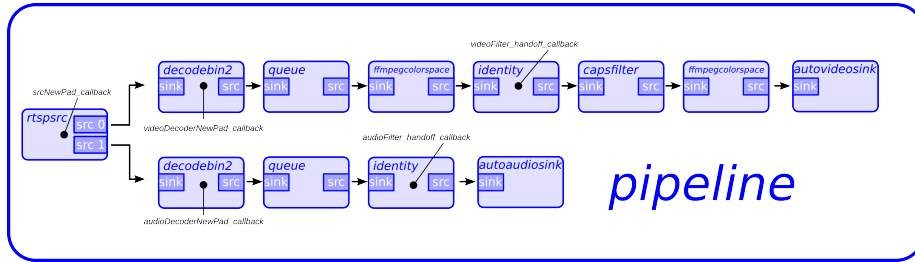


Figura 1: Estructura de la linea de flujo en rovioStreaming.

```
//-----
class ADT_rovioAPI
{
private:
    string uri;
    int sendCommand(const char *command) const;
    static int curlCallback(char *data, unsigned int size, unsigned int nmemb,
                           void *userdata);

public:
    rovioStreaming *streaming;
    rovioSpeak *speak;
    void setHome() const;
    void goHome() const;
    void stop() const;
    void move(double angle, unsigned int speed) const;
    void rotate(double angle, unsigned int speed) const;
    void headUp() const;
    void headMiddle() const;
    void headDown() const;
    void light(int l) const;
    const char* getURI() const;
    ADT_rovioAPI(const char *_uri);
    ~ADT_rovioAPI();
};
//-----
```

Figura 2: ADT_rovioAPI.h

3. Ejemplo

Junto con el código de esta API se provee de un programa ejemplo para demostrar su funcionamiento. El programa ejemplo consiste en una interfaz gráfica en GTK+ y glade con la cual se accesa a las funciones del API.

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. La interfaz gráfica se puede editar con el entrono de diseño Glade el cual es el paquete **glade-gnome**.

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.