PRÁCTICA 13

Java Multimedia Framework

Descripción de la práctica

El objetivo de esta práctica es realizar una aplicación que permita reproducir medios continuos, acceder a la cámara y capturar imágenes instantáneas usando la API del *Java Multimedia Framework (JMF)*. El aspecto visual de la aplicación será el mostrado en la Figura 1, donde el escritorio podrá tener tres tipos de ventanas[1]:

- Ventana de reproducción, que permitirá visualizar vídeos o escuchar audio (cada ventana irá asociada a un fichero de audio/vídeo). Esta ventana, además, incluirá un panel de control que permita parar y avanzar la reproducción.
- Ventana de cámara, que mostrará la captura realizada a través de la webcam.
- Ventana de imagen[2], que mostrará instantáneas capturadas de los vídeos/cámara mostrados en las ventanas anteriores.

En el menú se incluirá la opción "Archivo" que tendrá a su vez la opción "Abrir audio/video" La primera deberá lanzar el correspondiente diálogo y crear una nueva ventana interna de reproducción que permita reproducir el fichero abierto. Además, se incluirá una barra de herramientas que contenga dos botones: uno asociado a la opción "Cámara", que lanzará una ventana que muestre la secuencia que esté captando la webcam, y otro vinculado a la opción "Captura", que permitirá la captura de imágenes de la cámara o del vídeo.

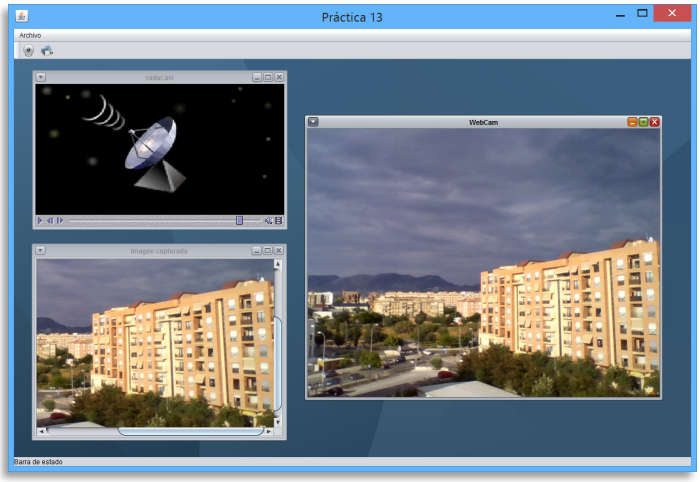

Figura 1: Aspecto de la aplicación

Instalación JMF

- En primer lugar, hay que instalar el JMF. Para ello, podemos optar por dos alternativas:

  a. Bajarnos el JMF de la página oficial de Java e instalarlo: http://www.oracle.com/technetwork/java/javase/download-142937.html

[1] En caso de continuar con la práctica 12, además de las ventanas incluidas en esta práctica, tendremos también las ventanas internas de reproducción y captura de audio. Recordemos que, cuando hay varios tipos de ventanas internas, hay que tenerlo en cuenta en la llamada al `getSelectedFrame()`.

[2] Esta ventana corresponde a la ventana interna desarrollada en las prácticas 8-11.

En este caso, la instalación incluirá los ficheros .jar asociados a el JMF[3], así como la aplicación JMStudio (demo que usa JMF y que es útil, entre otras cosas, para saber qué dispositivos de captura hay instalados y su nombre).

b. Bajarnos directamente de la web de la práctica los ficheros .jar (en este caso, no tendríamos la aplicación JMStudio)

Independientemente de la opción elegida, habrá que incorporar los ficheros .jar al proyecto NetBeans (a través de *Propiedades* → *Biblioteca* → *Añadir JAR*).

- Por otro lado, recordemos que el JMF no funciona correctamente con el JDK de 64 bits, por lo que deberemos de ejecutar nuestra aplicación con un JDK de 32 bits. En caso de que nuestro sistema sea de 64 bits, tendremos que:

  - Bajarnos el JDK de 32 bits e instalarlo en nuestro sistema (http://www.oracle.com/technetwork/java/javase/downloads/index.html?ssSourceSiteId=otnjp)

  - En el proyecto NetBeans, incluir la opción de ejecutar el proyecto usando el JDK de 32 bits recién instalado. Para ello, en la sección *Propiedades* → *Biblioteca* → *Plataforma Java*, seleccionaremos la opción de "Administrar plataformas" y añadiremos una nueva indicándole como ubicación del JDK la correspondiente al JDK de 32 bits instalado.

Ventana de reproducción

La ventana de reproducción permitirá reproducir tanto vídeo como sonido (los formatos y códecs soportados por JMF). La zona inferior de la ventana mostrará un panel de control y, en el caso del vídeo, la zona central mostrará el área de visualización.

Algunas recomendaciones:

- Crear una clase `VentanaInternaJMFPlayer` (que herede de `JInternalFrame`) que contenga una variable de tipo `Player`. El constructor de dicha clase creará el objeto `Player` asociado a la ventana, para lo cual se recomienda pasar el fichero a reproducir como parámetro del constructor:

```
private Player player = null;
```

```
private VentanaInternaJMFPlayer(File f) {
  initComponents();
  String sfichero = "file:" + f.getAbsolutePath();
  MediaLocator ml = new MediaLocator(sfichero);
  try {
    player = Manager.createRealizedPlayer(ml);
    Component vc = player.getVisualComponent();
    if(vc!=null)add(vc, java.awt.BorderLayout.CENTER);
    Component cpc = player.getControlPanelComponent();
    if(cpc!=null)add(cpc, java.awt.BorderLayout.SOUTH);
    this.pack();
  }catch(Exception e) {
    System.err.println("VentanaInternaJMFPlayer: "+e);
    player = null;
  }
}
```

La creación del `Player` puede generar excepciones que impliquen la no realización del mismo, en cuyo caso no debería de lanzarse la ventana. El uso de un constructor estándar implicaría siempre la creación de la ventana interna, por lo que se aconseja la definición de un método `getInstance(File f)` que llame internamente al constructor (que se declararía privado) y que, en caso de error en la creación del `Player`, devuelva `null`:

[3] En el caso del sistema windows, lo instala por defecto en "*C:\ Archivos de programa \JMF2.1.1e\lib*"

```
public static VentanaInternaJMFPlayer getInstance(File f){
  VentanaInternaJMFPlayer v = new VentanaInternaJMFPlayer(f);
  if(v.player!=null) return v;
  else return null;
}
```

- Definir los métodos `play()` y `close()` que lancen y detengan la reproducción respectivamente. Por ejemplo, para el caso del `play()`:

```
public void play() {
  if (player != null) {
    try {
      player.start();
    } catch (Exception e) {
      System.err.println("VentanaInternaJMFPlayer: "+e);
    }
  }
}
```

- Gestionar el evento "`InternalFrameClosing`" asociado al cierre de la ventana interna y llamar al método `close()`.

Ventana cámara[4]

La ventana de cámara mostrará la secuencia que esté captando la webcam (el área visual estará en el centro de la ventana, sin necesidad de panel de control en la parte inferior).

Algunas recomendaciones:

- Crear una clase `VentanaInternaCamara` (que herede de `JInternalFrame`) que contenga una variable de tipo `Player`. El constructor de dicha clase creará el objeto `Player` asociado a la cámara[5]. Al igual que en el caso anterior, se aconseja la definición de un método `getInstance()`.

- Al igual que en el caso de la ventana de reproducción, definir un método `close()` que cierre el `Player` y llamarlo desde el manejador asociado al evento "`InternalFrameClosing`" de la ventana interna.

Captura de instantáneas

Mediante el uso del botón situado en la barra de herramientas, el usuario podrá capturar imágenes de la cámara o del vídeo que se esté reproduciendo; concretamente, lo hará de la ventana que esté activa, siempre y cuando sea una ventana de tipo "reproducción" o "cámara". Para ello se usará el código visto en teoría (método `getFrame(Player)`). La imagen capturada será mostrada usando una ventana interna "imagen" como la diseñada en las prácticas 8-11[6].

---

[4] El acceso a la cámara web puede dar problemas, ya sea porque no identifica alguna de las cámaras instaladas (no saliendo en la lista de cámaras web disponibles) o porque, aun detectándolas, no puede conectarse (generando la excepción `java.io.IOException: Could not connect to capture device`). En este caso, se aconseja ejecutar el programa "*JMF Registry*", que se instala con el JMF, y comprobar que hay dispositivos de captura instalados (en caso negativo, darle al botón de detectar); también se aconseja ejecutar el programa "*JMStudio*" y probar si hay acceso a la cámara web (menú *File* → *Capture*).

[5] El código será similar al constructor de la clase `VentanaInternaJMFPlayer`, pero en este caso el localizador del medio tendrá que estar vinculado a la cámara web (véanse transparencias de teoría).

[6] Nótese que sobre esta imagen se podrían aplicar todas las operaciones desarrolladas en las prácticas 9-11.

Posibles mejoras para trabajar en casa…

En primer lugar, se propone como mejora el uso de una única opción “Abrir” que gestione la lectura tanto de imágenes como de audio y vídeo. En este caso, el filtro del diálogo incluiría tanto tipos de archivo de imágenes como de sonido y video.

Una mejora de mayor calado implicaría centralizar en una barra de herramientas las acciones de reproducción/pausa/parada; en particular, y si se incorporó la mejora propuesta en la práctica 12 en esta misma línea, la idea sería usar los mismos botones de aquella barra que, en función del tipo de ventana que hubiese seleccionada, se usarían para reproducir tanto audio como vídeo. A los botones ya existentes en aquella barra, le añadiríamos, además, los dos incorporados en ésta (captura e instantánea).