Graphical User Interfaces

QuickTime for Java

Presentation of the Semester Thesis' Toolkit
within the Context of
„Graphical User Interfaces“ (MSc.)

Department of Computer Science / Mathematics
Munich University of Applied Sciences

Table of Contents

- What is QuickTime?
- What is QuickTime for Java?
  - Introduction
  - Usage
  - Supported File Formats
  - Installation (Windows)
- How to use it?
  - Converting a Movie
  - Playing a Movie
- Examples
- Further Resources

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

What is QuickTime?

„QuickTime is a multimedia technology developed by Apple Computer, capable of handling various formats of digital video, sound, text, animation, music, and immersive panoramic (and sphere panoramic) images.“

(by en.wikipedia.org)

Key Features

- Cross-platform ability (Windows, Mac OS, certain hand-helds, etc.)
- Combination of authoring, delivery and playback
- Supports dozens of different file formats, video and audio codecs
- Use QuickTime features in your own programs (C++, Java, etc.)

Capabilities

- Displaying media (videos, images, audio)
- Media streaming
- Conversion between different formats
- Video visual effects
- etc.

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Table of Contents

- What is QuickTime?
- What is QuickTime for Java?
  - Introduction
  - Usage
  - Supported File Formats
  - Installation (Windows)
- How to use it?
  - Converting a Movie
  - Playing a Movie
- Examples
- Further Resources

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

What is QuickTime for Java?

Basics

- Java Application Programming Interface (API)
- Developed and maintained by Apple
- Provides access to all QuickTime features
- Runs on all operating systems offering a QuickTime implementation

Supported Operating Systems

- Windows 98, ME, 2000, XP, 2003
- Mac OS 8.x, Mac OS 9.x, Mac OS, 10.x

Drawbacks

- No Linux support (despite some projects running QuickTime on Linux)
- Closed-source product

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Usage

Java-specific

- Applets
- Web Start applications
- Stand-alone applications
- Server-side applications (Java Servlets)

Capabilities cont.

- Displaying (videos, images, audio)
- Conversion (file import, export)
- Capturing (recording from video sources)
- Streaming (receiving, broadcasting)
- Editing (author videos, images, audio)
- Effects (create video effects)
- Manipulation (pixel-wise editing of videos, images)

Supported File Formats [7],[8]

Import Formats

3DMF (Mac OS 9 & Windows), 3GPP, 3GPP2, AIFF, AMC, AMR, Animated GIF, AU, Audio CD Data (Mac OS 9), AVI, BMP, CAF (Mac OS X), Cubic VR, DLS, DV, FlashPix, FLC, GIF, GSM, JPEG 2000, JPEG/JFIF, Karaoke, MacPaint, Macromedia Flash 5, MIDI, MPEG-1, MP3(MPEG-1, Layer 3), M3U(MP3 Playlist files), MPEG-2, MPEG-4, MQV, M4A, M4B, M4P (iTunes 4 audio), M4V (iTunes video), PDF (Mac OS X), Photoshop, PICS, PICT, PLS, PNG, Quartz Composer Composition (Mac OS X), QCP (Mac OS 9 & Windows), QuickTime Image File, QuickTime Movie, SD2 (Mac OS 9 & Windows), SDP, SDV, SF2 (SoundFont 2), SGI, SMIL, System 7 Sound (Mac OS 9), Targa, Text, TIFF, TIFF Fax, VDU (Sony Video Disk Unit), Virtual Reality (VR), Wave

Export Formats

3GPP, 3GPP2, AIFF, AMC, AU, AVI, BMP, DV Stream, FLC, Image Sequence movie exporters, JPEG/JFIF, JPEG 2000, MacPaint, MIDI, MPEG-4, Photoshop, PICT, PNG, QuickTime Image File, QuickTime Movie, SGI, System 7 Sound, Targa, Text, TIFF, WAV

Video Codecs

Animation, Apple BMP, Apple Pixlet (Mac OS X v10.3 only), Apple Video, Cinepak, Component video, DV and DVC Pro NTSC, DV PAL, DVC Pro PAL, Graphics, H.261, H.263, H.264, JPEG 2000, Microsoft OLE (decode only), Microsoft Video 1 (decode only), Motion JPEG A, Motion JPEG B, MPEG-4 (Part 2), Photo JPEG, Planar RGB, PNG, Sorenson Video 2, Sorenson Video 3, TGA, TIFF

Audio Codecs

24-bit integer, 32-bit floating point, 32-bit integer, 64-bit floating point, AAC (MPEG-4 Audio), ALaw 2:1, AMR Narrowband, Apple Lossless Encoder, IMA 4:1, MACE 3:1, MACE 6:1, MS ADPCM (decode only), QDesign Music 2, Qualcomm PureVoice (QCELP), ULaw 2:1

Installation (Windows)

Requirements

- QuickTime (latest v7)
- QuickTime for Java (must be selected during „Custom“ installation)
- Sun Java JRE/JDK (latest v5)

Optional Requirements

- IDE of your choice (i.e. Eclipse)
- QuickTime for Java SDK (JavaDoc, sample code)

Steps

- Install Java JRE/JDK
- Install QuickTime (remember to selected the Java option)
- Add „QTJava.zip“ to the compiler's classpath
- Add directory containing „QTJava.dll“ and „QTJavaNative.dll“ to the Java VM arguments: -Djava.library.path=/path/to/libs/

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Table of Contents

- What is QuickTime?
- What is QuickTime for Java?
  - Introduction
  - Usage
  - Supported File Formats
  - Installation (Windows)
- How to use it?
  - Converting a Movie
  - Playing a Movie
- Examples
- Further Resources

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Converting a Movie

```
/* MovieConverter.java
 * for transmediale QTJ workshop, February 2004
 * copyright (c) 2004 by Jason Freeman
 * relased under terms of the GNU public license
 * http://www.gnu.org/licenses/licenses.html#TOCGPL
 *
 * some portions based on work by:
 * (c) 2003, Chris Adamson, invalidname@mac.com
 * relased under terms of the GNU public license
 * http://www.gnu.org/licenses/licenses.html#TOCGPL
 *
 *
 * ABOUT THIS PROGRAM:
 * MovieConverter presents a dialog box allowing the user to select any media file recognized by Quicktime.
 * It then presents a second dialog box allowing the user to convert the media file to any format supported
 * by Quicktime and saves the converted file as a new file.*/

public class MovieConverter extends Frame {

        Movie movie; // Basic Quicktime class representing all types of media

        public MovieConverter(String title) {
                super(title);

                // Place the working code here ...
        }

        public static void main(String[] args) {
                MovieConverter frame = new MovieConverter("QTJ Movie Converter");
                System.exit(0);
        }
}
```

Converting a Movie cont.

```
[...]

try {
        QTSession.open(); // Initialize the Quicktime engine

        FileDialog fd = new FileDialog(this, "Select source movie", FileDialog.LOAD);
        fd.setVisible(true);
        if (fd.getFile() == null) return;
        File f = new File(fd.getDirectory(), fd.getFile());

        // Open the media file with Quicktime
        QTFile inputFile = new QTFile(f);
        OpenMovieFile omFile = OpenMovieFile.asRead(inputFile);
        movie = Movie.fromFile(omFile);

        // Show progress bar if conversion operation takes a long time
        movie.setProgressProc();

        // Convert the movie into a new format and save it as a new file
        //int exportType = StdQTConstants.kQTFileTypeMovie;   // Limit to exporting Quicktime Movies
        int exportType = 0;

        movie.convertToFile(new QTFile("Export Movie"),         // Temp. name of movie which shows up in dialog box
                        exportType,                             // Allow conversion to any supported file type
                        StdQTConstants.kMoviePlayer,            // Set the creator of the file to be the MoviePlayer
                        IOConstants.smSystemScript,             // What script movie should be converted into
                        StdQTConstants.showUserSettingsDialog); // Show a dialog box to let user configure the details

        // Shut down the Quicktime engine
        QTSession.close();

} catch (Exception e) {
        e.printStackTrace();
}

[...]
```


Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Playing a Movie

```
/* ExamplePlayer.java
 * by Thomas Wöllert, 2006
 *
 * ABOUT THIS PROGRAM:
 * ExamplePlayer presents a dialog box allowing the user to select any media file recognized by Quicktime.
 * It then opens a new frame to show the movie. A control bar is also added to start and stop the movie
 * playback. Also a scrollbar enables the user to jump to a certain point within the movie. */

public class ExamplePlayer extends Frame {

	Movie movie; // Basic Quicktime class representing all types of media

	public ExamplePlayer(String title) {
		super(title);

		// Place the working code here ...
	}

	public static void main(String[] args) {
		ExamplePlayer frame = new ExamplePlayer("QTJ Movie Player");
	}
}
```


Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Playing a Movie cont.

```
[...]

try {
        QTSession.open(); // Initialize the Quicktime engine

        FileDialog fd = new FileDialog(this, "Select source movie", FileDialog.LOAD);
        fd.setVisible(true);
        if (fd.getFile() == null) return;
        File f = new File(fd.getDirectory(), fd.getFile());

        // Open the media file with Quicktime
        QTFile inputFile = new QTFile(f);
        OpenMovieFile omFile = OpenMovieFile.asRead(inputFile);
        movie = Movie.fromFile(omFile);

        // Create the movie controller (which also acts as player)
        MovieController movCont = new MovieController(movie);

        // Create the controller component
        QTComponent qtc = QTFactory.makeQTComponent(movCont);
        Component c = qtc.asComponent();

        // Add the controller to the frame and show it
        this.add(c);
        this.pack();
        this.setVisible(true);

        // Closing the QTSession is not shown in this example

} catch (Exception e) {
        e.printStackTrace();
}

[...]
```


Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Table of Contents

- What is QuickTime?
- What is QuickTime for Java?
  - Introduction
  - Usage
  - Supported File Formats
  - Installation (Windows)
- How to use it?
  - Converting a Movie
  - Playing a Movie
- Examples
- Further Resources

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Examples

Taken from the QuickTime for Java SDK [5] ...

AddTextMovie
Shows how to add a text track to a movie.

AudioBroadcaster
Shows how to use the broadcasting API to broadcast audio from a Java application.

CurvesDemo
Shows how to use the curve codec in QuickTime to draw a vector graphic image.

JavaRadioStation
RadioStation is a small but functional streaming MP3 server written in pure Java. In other words, it lets you broadcast high-quality MPEG audio across the Internet or a LAN, just like ShoutCast or Live365. The source code provides a (hopefully) clearly-written example of a multithreaded networked server, with some nontrivial producer/consumer synchronization and bandwidth management, which uses Java file I/O to parse the fairly complex MPEG and ID3 data formats and do some macro substitution on HTML.

MediaPresenter
Shows how to rotate and scale any QuickTime presented content and keep that content postioned at top, left of display area.

SoundRecord
This code shows using the SequenceGrabber for simple audio recording.

Further Resources (as of May 2006)

Apple – Basic Resources

[1] *QuickTime Java - Main,* http://developer.apple.com/java/

[2] *QuickTime for Java*, http://developer.apple.com/quicktime/qtjava/

[3] *QuickTime for Java API*, http://developer.apple.com/quicktime/qtjava/javadocs.html

[4] *QuickTime SDK Download*, http://developer.apple.com/sdk/index.html#QTJava

[5] *QuickTime Java Sample Code*, http://developer.apple.com/samplecode/QuickTime/idxJava-date.html

[6] *An Introduction to QuickTime for Java,* http://developer.apple.com/quicktime/qtjava/qtjtutorial/

[7] *Supported File Formats,* http://www.apple.com/quicktime/player/specs.html

[8] *QuickTime Pro – Tech Specs,* http://www.apple.com/quicktime/pro/specs.html

Apple – Advanced Resources

[9] *Built-In QuickTime Video Effects*,
http://developer.apple.com/documentation/QuickTime/REF/refEffects.3d.htm#pgfId=324

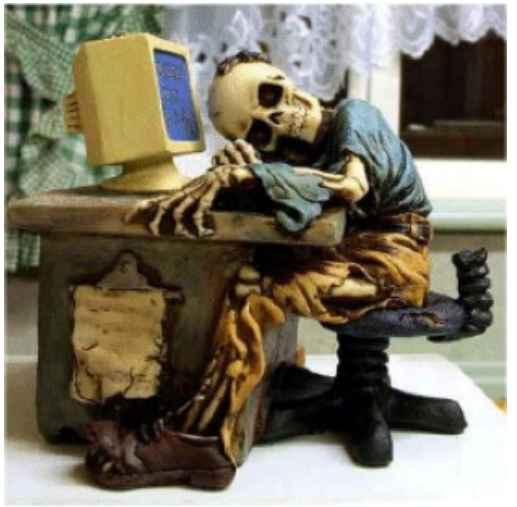
[10] *Interactive Movies*,
http://developer.apple.com/documentation/quicktime/IQ_InteractiveMovies/index.html

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Ende – End – Fin - Conclude

Thank you for your attention !

Questions ?



Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)