Multimedia
BSc Exam Questions Jan 2001 SOLUTIONS

Exam paper format:

(b) Time Allowed: 2 Hours
(c) Answer 3 Questions out of 4
(d) Each Question Carries 27 Marks

1. (a) What is meant by the terms Multimedia and Hypermedia? Distinguish between these two concepts.

Multimedia ---- An Application which uses a collection of multiple media sources e.g. text, graphics, images, sound/audio, animation and/or video.

Hypermedia --- An application which uses associative relationships among information contained within multiple media data for the purpose of facilitating access to, and manipulation of, the information encapsulated by the data.

2 MARKS ---- BOOKWORK

(b) What is meant by the terms static media and dynamic media? Give two examples of each type of media.

Static Media – does not change over time, e.g. text, graphics

Dynamic Media --- Time dependent (Temporal), e.g. Video, sound, animation.

4 MARKS --- BOOKWORK

(c) What are the main facilities that must be provides in a system designed to support the integration of multimedia into a multimedia presentation?

The following functionality should be provided:

- Digital Representation of Media --- Many formats for many media
- Capture: Digitisation of Media --- special Hardware/Software

- Creation and editing --- assemble media and alter it
- Storage Requirements --- significant for multimedia
- Compression --- related to above and below, ie can save on storage but can hinder retrieval
- Structuring and retrieval methods of media --- simple to advanced DataBase Storage
- Display or Playback methods --- effect of retrieval must view data
- Media Synchronisation --- display multimedia as it is intended

8 MARKS --- BOOKWORK

(*d) Describe giving suitable code fragments how you would effectively combine and start at the same time a video clip and an audio clip in an MHEG application and start a subtiltle text display 19000 milliseconds into the video clip. You may assume that both clips are of the same duration and must start at the same instant.*

TheMHEG code listing below is illustrates the solution:

```
{:Application ("SYNC_APP.mh5" 0)
:OnStartUp ( // sequence of initialization
actions
}
{:Scene ("main_scene.mh5" 0)
:OnStartUp ( // sequence of initialization
actions
preload (2) // the connection to the
source of the video clip is set up
preload (3) // the connection to the
source of the audio clip is set up

...
setCounterTrigger (2 3 190000) // book a
time code event at 190000 msec for example
...
)
:Items ( // both presentable ingredients and
links
{:Bitmap 1 // background bitmap  NOT IMPORTANT FOR SOLN
:InitiallyActive true
:CHook 3 // JPEG
:OrigContent
:ContentRef ("background.jpg")
:OrigBoxSize 800 600
:OrigPosition 0 0
}
{:Stream 2 // video clip
:InitiallyActive false
:CHook 101 // MPEG-1
:OrigContent
:ContentRef ("video.mpg")
:Multiplex (
{:Audio 3 // audio component of the
video clip
:ComponentTag 1 // refers to audio
elementary stream
:InitiallyActive true
}
... // possibly  more presentable ingredients

}
{:Link 49 // the video clip crosses a pre
defined time code position
:EventSource (2) // video clip
:EventType CounterTrigger
:EventData 3 // booked at startup by
setCounterTrigger (2 3 190000)
:LinkEffect (
:SetData (5 // text subtitle is set
to a new string, that is
:NewRefContent ("subtitle.txt")) //
"Subtitle text"
:SetHighlightStatus (20 true) //
hotspot 20 is highlighted
)
}
... //  more links
)
:SceneCS 800 600 // size of the scene's
presentation space
}
```

Key Points:

- Preloading both clips is essential to start streaming:
- Need to “book” 190000 msec event for subtitles
- Content loaded and Video and audio is multiplexed
- Set a link transition for subtiltle

13 MARKS - UNSEEN

2. (a) Why is file or data compression necessary for Multimedia activities?

Multimedia files are very large therefore for storage, file transfer etc. file sizes need to be reduced often as part of the file format. Text and other files may also be encoded/compressed for email and other applications.

3 Marks -- Bookwork

(b) Briefly explain how the LZW Transform Operates. What common compression methods utilise this transform.

Suppose we want to encode the Oxford Concise English dictionary which contains about 159,000 entries. Why not just transmit each word as an 18 bit number?

Problems:

* Too many bits,
* everyone needs a dictionary,
* only works for English text.
* **Solution**: Find a way to build the dictionary adaptively.

- Original methods due to Ziv and Lempel in 1977 and 1978. Terry Welch improved the scheme in 1984 (called LZW compression).
- It is used in UNIX *compress* and GIF compression

The LZW Compression Algorithm can summarised as follows:

```
w = NIL;
while ( read a character k )
    {
      if wk exists in the dictionary
       w = wk;
      else
        add wk to the dictionary;
        output the code for w;
        w = k;
    }
```

* Original LZW used dictionary with 4K entries, first 256 (0-255) are ASCII codes.

10 MARKS – BOOKWORK

(c) Show how the LZW transform would be used to encode the following 2D array of image data, you should use 2x2 window elements for the characters:

0	1	0	2	2	0
2	0	1	1	1	0
0	2	0	1	0	2
1	0	2	0	1	1
2	0	0	2	0	2
1	0	1	0	1	1

SEE HANDWRITTEN SOLN attached

14 Marks UNSEEN

3 (a) What key features of Quicktime have lead to its adoption and an international multimedia format?

QuickTime is the most widely used cross-platform multimedia technology available today. QuickTime developed out of a multimedia extension for Apple's Macintosh(proprietry) System 7 operating system. It is now an international standard for multimedia interchange and is avalailbe for many platforms and as Web browser plug ins.

The following main features are:

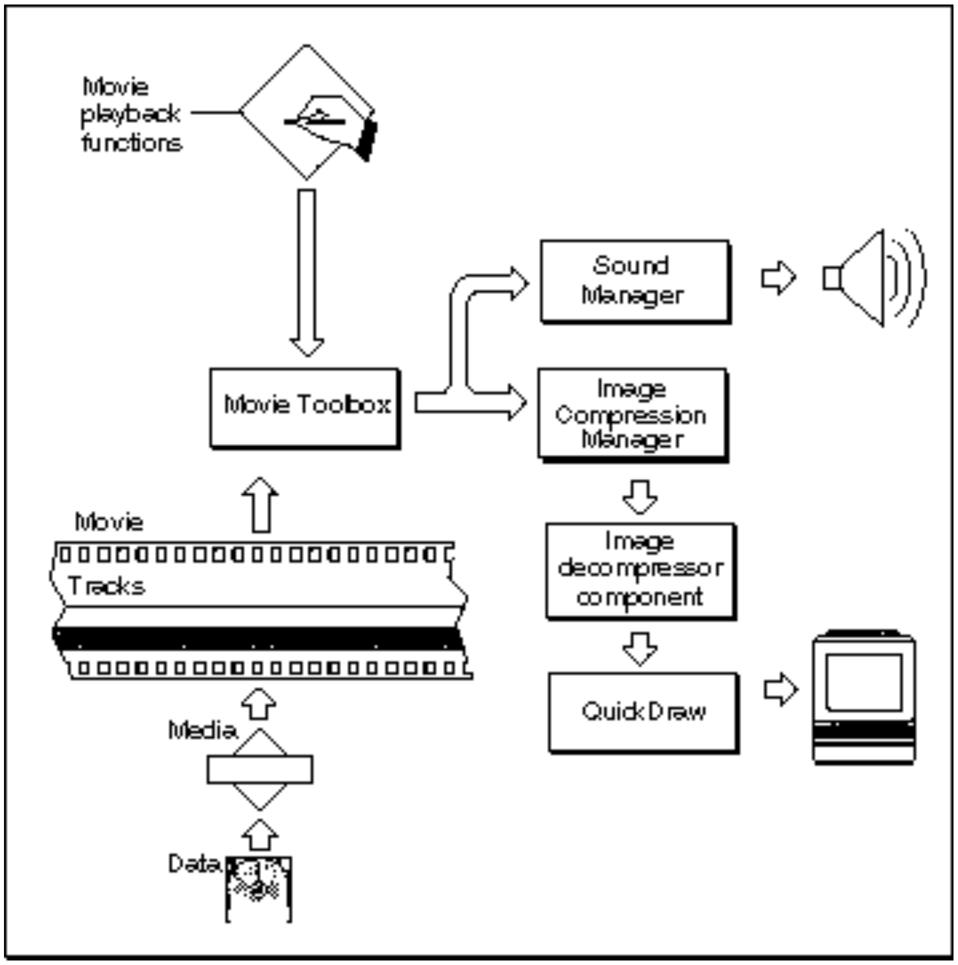
- Versatile support for web-based media
- Sophisticated playback capabilities
- Easy content authoring and editing
- QuickTime is an ***open standard*** -- it embraces other standards and incorporates them into its environment. It supports almost every major Multimedia file format

4 Marks – BOOKWORK

(b) Briefly outline the Quicktime Architecture and its key components.

The QuickTime Architecture:

QuickTime comprises two managers: the Movie Toolbox and the Image Compression Manager. QuickTime also relies on the Component Manager, as well as a set of predefined components. Figure below shows the relationships of these managers and an application that is playing a movie.



The Movie Toolbox
-- Your application gains access to the capabilities of QuickTime by calling functions in the Movie Toolbox. The Movie Toolbox allows you to store, retrieve, and manipulate time-based data that is stored in QuickTime movies. A single movie may contain several types of data. For example, a movie that contains video information might include both video data and the sound data that accompanies the video.

The Movie Toolbox also provides functions for editing movies. For example, there are editing functions for shortening a movie by removing portions of the video and sound tracks, and there are functions for extending it with the addition of new data from other QuickTime movies.

The Movie Toolbox is described in the chapter "Movie Toolbox" later in this book. That chapter includes code samples that show how to play movies.

The Image Compression Manager
--

The Image Compression Manager comprises a set of functions that compress and decompress images or sequences of graphic images.

The Image Compression Manager provides a device-independent and driver-independent means of compressing and decompressing images and sequences of images. It also contains a simple interface for implementing software and hardware image-compression algorithms. It provides system integration functions for storing compressed images as part of PICT files, and it offers the ability to automatically decompress compressed PICT files on any QuickTime-capable Macintosh computer.

In most cases, applications use the Image Compression Manager indirectly, by calling Movie Toolbox functions or by displaying a compressed picture. However, if your application compresses images or makes movies with compressed images, you will call Image Compression Manager functions.

The Image Compression Manager is described in the chapter "Image Compression Manager" later in this book. This chapter also includes code samples that show how to compress images or make movies with compressed images.

The Component Manager
--

Applications gain access to components by calling the Component Manager. The Component Manager allows you to define and register types of components and communicate with components using a standard interface. A component is a code resource that is registered by the Component Manager. The component's code can be stored in a systemwide resource or in a resource that is local to a particular application.

> Once an application has connected to a component, it calls that component directly. If you create your own component class, you define the function-level interface for the component type that you have defined, and all components of that type must support the interface and adhere to those definitions. In this manner, an application can freely choose among components of a given type with absolute confidence that each will work.

QuickTime Components :

- movie controller components, which allow applications to play movies using a standard user interface standard image compression dialog components, which allow the user to specify the parameters for a compression operation by supplying a dialog box or a similar mechanism
- image compressor components, which compress and decompress image data sequence grabber components, which allow applications to preview and record

video and sound data as QuickTime movies video digitizer components, which allow applications to control video digitization by an external device

- media data-exchange components, which allow applications to move various types of data in and out of a QuickTime movie derived media handler components, which allow QuickTime to support new types of data in QuickTime movies
- clock components, which provide timing services defined for QuickTime applications preview components, which are used by the Movie Toolbox's standard file preview functions to display and create visual previews for files sequence grabber components, which allow applications to obtain digitized data from sources that are external to a Macintosh computer
- sequence grabber channel components, which manipulate captured data for a sequence grabber component
- sequence grabber panel components, which allow sequence grabber components to obtain configuration information from the user for a particular sequence grabber channel component

10 Marks BookWork

(c) Quicktime provides many basic built-in visual effect procedures. By using fragments of Java code show how a cross-fade effect between two images can be created. You solution should concentrate only on the Java code specific to producing the Quicktime effect. You may assume that the images are already imported into the application and are referred to as sourceImage and destImage. You should not consider any Graphical Interfacing aspects of the coding.

```
This code shows how a Cross Fade Transition effect could be built  NOT
ALL THE INTERFACING STUFF INCLUDED BELOW IS REQUIRED SEE COMMENTS AFTER
FOR IMPORTANT PARTS THAT NEED ADDRESSING.

/*
 * QuickTime for Java Transition Sample Code

 */

import java.awt.*;
import java.awt.event.*;
import java.io.*;

import quicktime.std.StdQTConstants;
import quicktime.*;
import quicktime.qd.*;
import quicktime.io.*;
import quicktime.std.image.*;
import quicktime.std.movies.*;
import quicktime.util.*;
import quicktime.app.QTFactory;
import quicktime.app.time.*;
import quicktime.app.image.*;
import quicktime.app.display.*;
import quicktime.app.anim.*;
import quicktime.app.players.*;
import quicktime.app.spaces.*;
import quicktime.app.actions.*;


public class TransitionEffect extends Frame implements
StdQTConstants, QDConstants {

     public static void main(String args[]) {
          try {
               QTSession.open();

               TransitionEffect te = new
TransitionEffect("Transition Effect");
               te.pack();
               te.show();
               te.toFront();
          } catch (Exception e) {
               e.printStackTrace();
               QTSession.close();
          }

     }



     TransitionEffect(String title) throws Exception {
          super (title);

          QTCanvas myQTCanvas = new
QTCanvas(QTCanvas.kInitialSize, 0.5f, 0.5f);
          add("Center", myQTCanvas);

          Dimension d = new Dimension (300, 300);
          addWindowListener(new WindowAdapter() {
               public void windowClosing (WindowEvent e) {
                    QTSession.close();
                    dispose();
               }

               public void windowClosed (WindowEvent e) {
                    System.exit(0);
               }
          });

          QDGraphics gw = new QDGraphics (new QDRect(d));
          Compositor comp = new Compositor (gw,
QDColor.black, 20, 1);

          ImagePresenter idb = makeImagePresenter (new
QTFile (QTFactory.findAbsolutePath ("pics/stars.jpg")),
                                             new
QDRect(300, 220));
          idb.setLocation (0, 0);
          comp.addMember (idb, 2);

          ImagePresenter id = makeImagePresenter (new
QTFile (QTFactory.findAbsolutePath ("pics/water.pct")),
                                             new
QDRect(300, 80));
          id.setLocation (0, 220);
          comp.addMember (id, 4);

          CompositableEffect ce = new CompositableEffect
();
          ce.setTime(800); // TIME OF EFFECT
           ce.setSourceImage(sourceImage);
           ce. setDestinationImage(destImage);
          ce.setEffect (createSMPTEEffect,
kEffectCrossFade, KRandomCrossFadeTransitionType);

          ce.setDisplayBounds (new QDRect(0, 220, 300,
80));
          comp.addMember (ce, 3);

          Fader fader = new Fader();
          QTEffectPresenter efp = fader.makePresenter();
          efp.setGraphicsMode (new GraphicsMode (blend,
QDColor.gray));
          efp.setLocation(80, 80);
          comp.addMember (efp, 1);

          comp.addController(new TransitionControl (20, 1,
fader.getTransition()));

          myQTCanvas.setClient (comp, true);
          comp.getTimer().setRate(1);
     }

     private ImagePresenter makeImagePresenter (QTFile
file, QDRect size) throws Exception {
```

```
          GraphicsImporterDrawer if1 = new
GraphicsImporterDrawer (file);
          if1.setDisplayBounds (size);
          return ImagePresenter.fromGraphicsImporterDrawer
(if1);
     }
}
```

……..

FULL CODE NOT REQUIRED as above

Important bits

- Set up an atom container to use an SMPTE effect using CreateSMPTEeffect()

- Set up a Transition with the IMPORTANT parameters:

    - ce.setTime(800); // TIME OF EFFECT
    - ce.setSourceImage(sourceImage);
    - ce. setDestinationImage(destImage);
    - ce.setEffect (createSMPTEEffect, kEffectCrossFade, KRandomCrossFadeTransitionType);

- A doTransition() or doAction() method performs transition

13 Marks --- UNSEEN

4. (a) *What is MIDI? How is a basic MIDI message structured?*

MIDI: a protocol that enables computer, synthesizers, keyboards, and other musical or (even) multimedia devices to communicate with each other.

MIDI MESSAGE:

- MIDI message includes a status byte and up to two data bytes.
- Status byte

  The most significant bit of status byte is set to 1.
- The 4 low-order bits identify which channel it belongs to (four bits produce 16 possible channels).
- The 3 remaining bits identify the message.
- The most significant bit of data byte is set to 0.

6 Marks --- Bookwork

(b) *In what ways can MIDI be used effectively in Multimedia Applications, as opposed to strictly musical applications*?

Many Application:

Low Bandwidth/(Low Quality?) Music on Web, Quicktime etc supports Midi musical instrument set
Sound Effectts --- Low Bandwidth alternative to audio samples, Sound Set part of GM soundset
Control of external devices --- e.g Synchronistaion of Video and Audio (SMPTE), Midi System Exclusive, AUDIO RECORDERS, SAMPLERS
Control of synthesis --- envelope control etc
MPEG 4 Compression control --- see Part (c)
Digital Audio

8 Marks --- Applied Bookwork: Discussion of Information mentioned in Notes/Lectures.

(c) How can MIDI be used with modern data compression techniques? Briefly describe how such compression techniques may be implemented?

- We have seen the need for compression already in Digital Audio -- Large Data Files

- Basic Ideas of compression (see next Chapter) used as integral part of audio format -- MP3, real audio etc.

- Mpeg-4 audio -- actually combines compression synthesis and midi to have a massive impact on compression.

- Midi, Synthesis encode what note to play and how to play it with a small number of parameters -- Much greater reduction than simply having some encoded bits of audio.

- Responsibility to create audio delegated to generation side.

MPEG-4 comprises of 6 Structured Audio tools are:

- SAOL the Structured Audio Orchestra Language

- SASL the Structured Audio Score Language
- SASBF the Structured Audio Sample Bank Format
- a set of MIDI semantics which describes how to control SAOL with MIDI
- a scheduler which describes how to take the above parts and create sound
- the AudioBIFS part of BIFS, which lets you make audio soundtracks in MPEG-4 using a variety of tools and effects-processing techniques

MIDI IS the control language for the synthesis part:

- As well as controlling synthesis with SASL scripts, it can be controlled with MIDI files and scores in MPEG-4. MIDI is today's most commonly used representation for music score data, and many sophisticated authoring tools (such as sequencers) work with MIDI.

- The MIDI syntax is external to the MPEG-4 Structured Audio standard; only references to the MIDI Manufacturers Association's definition in the standard. But in order to make the MIDI controls work right in the MPEG context, some semantics (what the instructions "mean") have been redefined in MPEG-4. The new semantics are carefully defined as part of the MPEG-4 specification.

  13 Marks --- UNSEEN but Basic Ideas mentioned in lectures and earmarked for further reading . Detailed application of Lecture notes material