ECE160
Multimedia

Lecture 2: Spring 2011
Multimedia Authoring and Tools

Structure

- Lectures: Tuesday and Thursday, 3:30pm-4:45pm
- Discussion/Lab: Tuesday, 5pm – 5:50pm
  Thursday, 9am – 9:50pm
- Teaching assistants:
  Abhinav Anand (abhinav_anand@umail.ucsb.edu),
  Nitin Chhabra (nitinchhabra@umail.ucsb.edu)

  TA Office Hours:

  Abhinav Anand: Wednesday, 1:00-4:00pm, ECI Lab
  Nitin Chhabra: Tuesday, 2:00-5:00pm, ECI Lab
- Web site for **ECE160**, with lectures and assignments

Multimedia Authoring and Tools

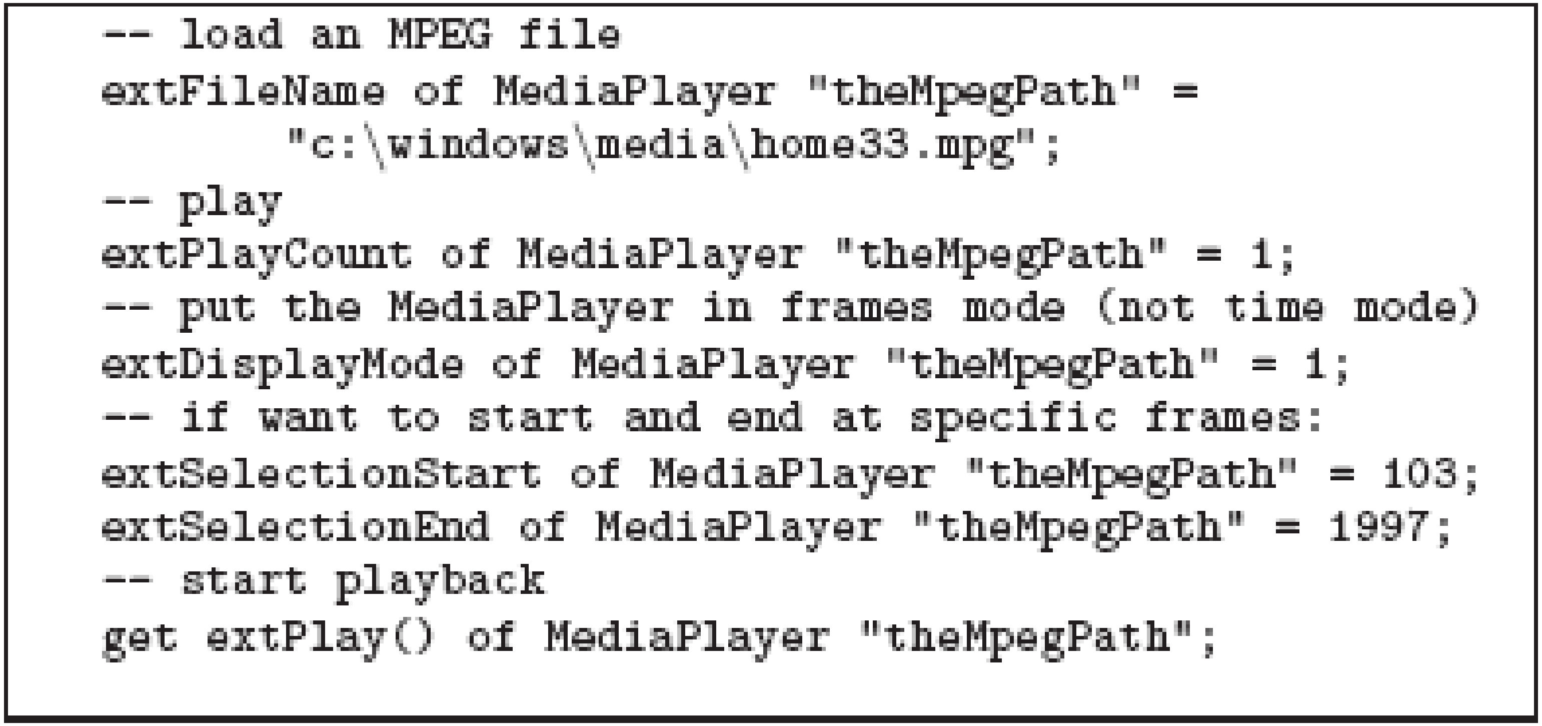
- Multimedia Authoring
- Some Useful Editing and Authoring Tools
- VRML

Multimedia Authoring

- **Multimedia authoring**: creation of multimedia productions, sometimes called “videos" or “presentations".
  - we are interested in **interactive** applications.
  - we also look at still-image editors such as Adobe Photoshop, and simple video editors such as Adobe Premiere.
- In this section, we take a look at:
  - **Multimedia Authoring Metaphors**
  - **Multimedia Production**
  - **Multimedia Presentation**
  - **Automatic Authoring**

Multimedia Authoring Metaphors

1. **Scripting Language Metaphor**: use a special language to enable interactivity (buttons, mouse, etc.), and to allow conditionals, jumps, loops, functions/macros etc.



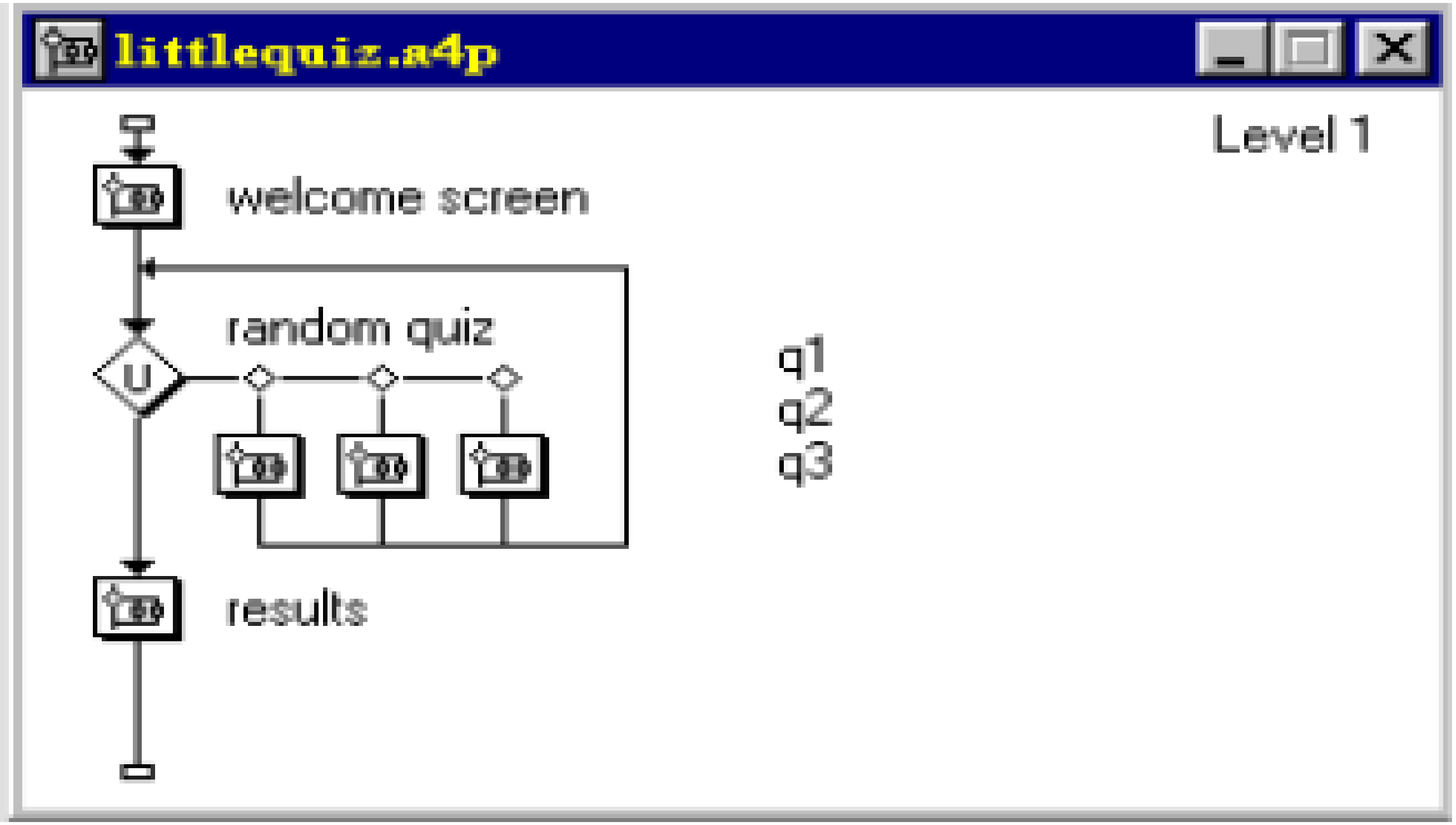
```
-- load an MPEG file
extFileName of MediaPlayer "theMpegPath" =
       "c:\windows\media\home33.mpg";
-- play
extPlayCount of MediaPlayer "theMpegPath" = 1;
-- put the MediaPlayer in frames mode (not time mode)
extDisplayMode of MediaPlayer "theMpegPath" = 1;
-- if want to start and end at specific frames:
extSelectionStart of MediaPlayer "theMpegPath" = 103;
extSelectionEnd of MediaPlayer "theMpegPath" = 1997;
-- start playback
get extPlay() of MediaPlayer "theMpegPath";
```

Multimedia Authoring Metaphors

2. **Slide Show Metaphor**: A linear presentation by default, although tools exist to perform jumps in slide shows.
3. **Hierarchical Metaphor**: User-controllable elements are organized into a tree structure - often used in menu-driven applications.
4. **Iconic/Flow-control Metaphor**: Graphical icons are available in a toolbox, and authoring proceeds by creating a flow chart with icons attached

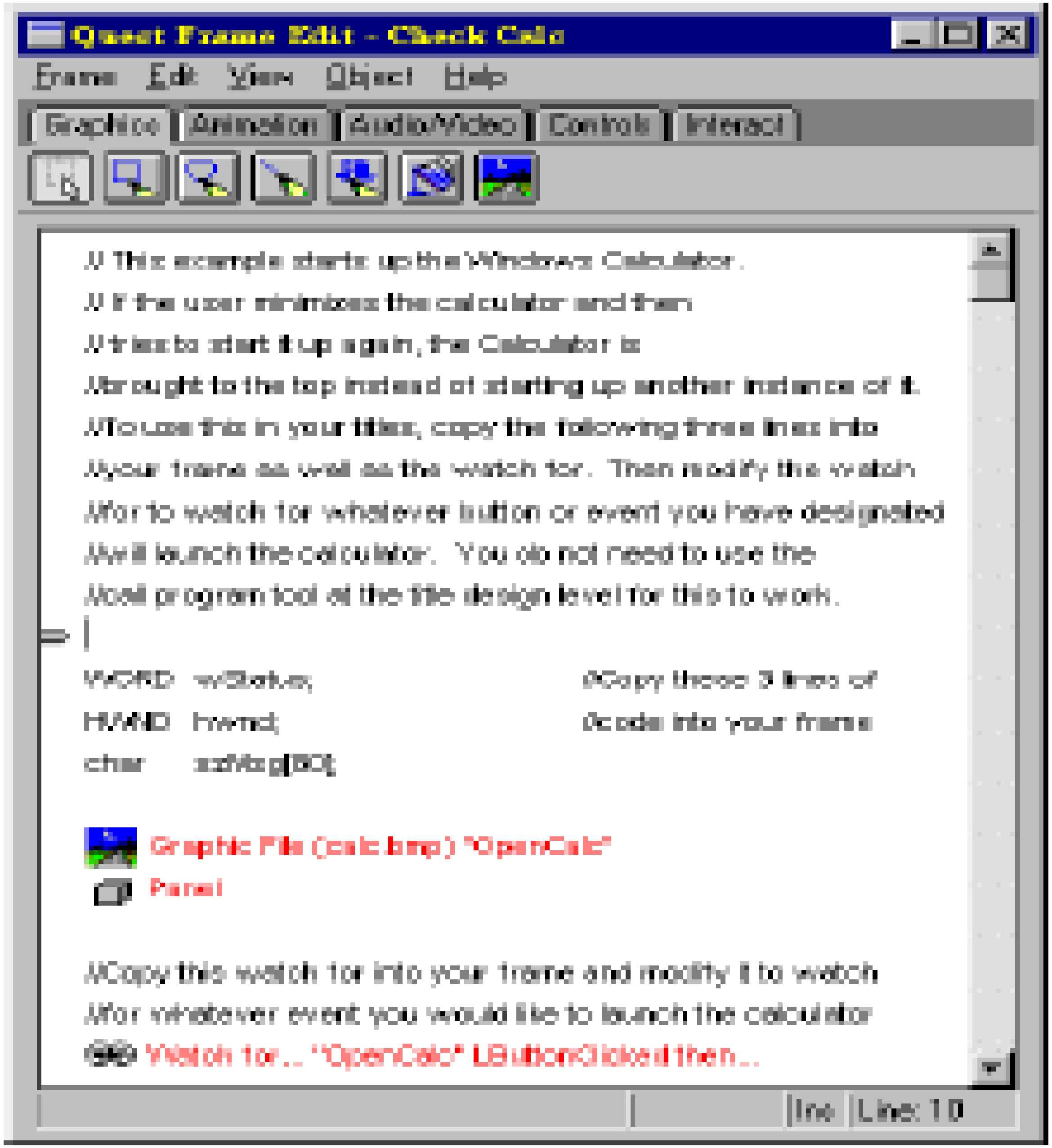
Multimedia Authoring Metaphors

- Authorware flowchart



Multimedia Authoring Metaphors

5. **Frames Metaphor**: Like Iconic/ Flow-control Metaphor; however links between icons are more conceptual, rather than representing the actual flow of the program

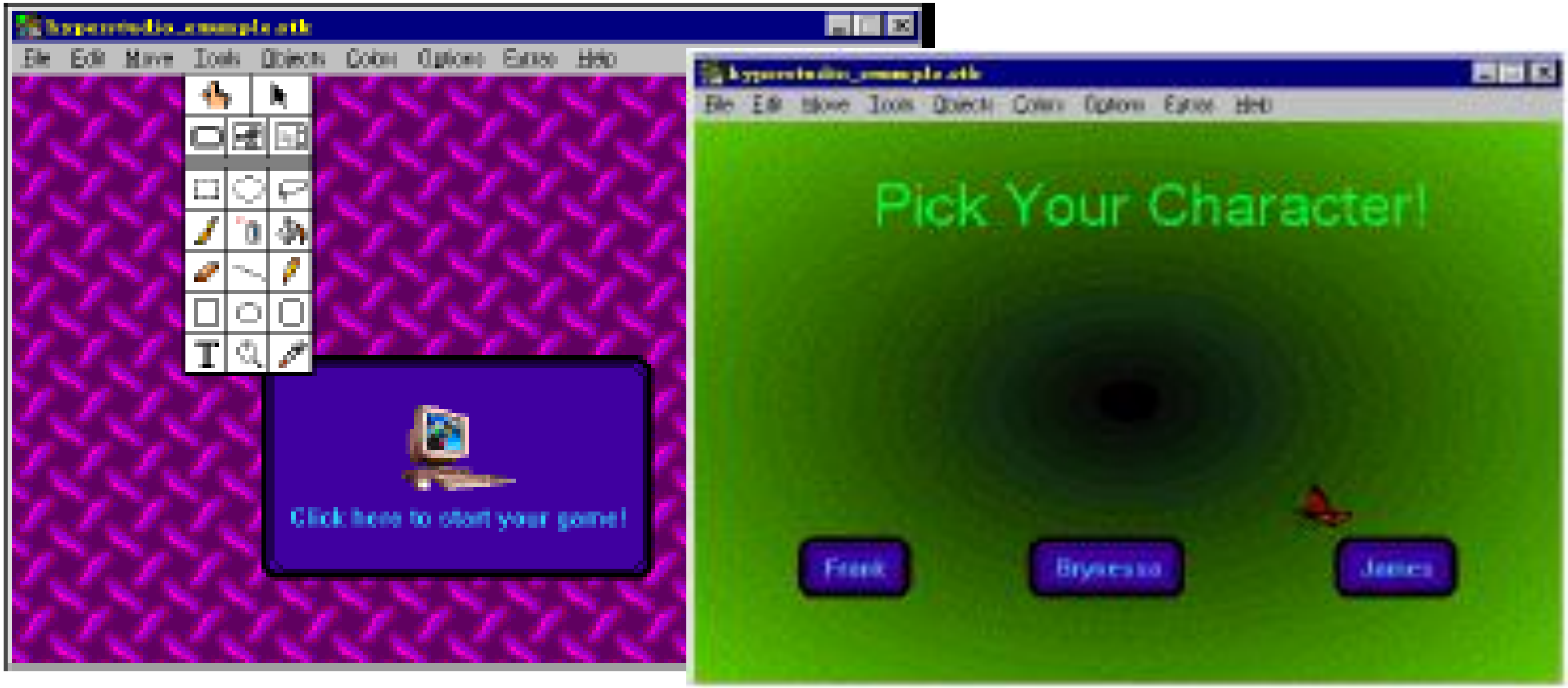


```
// This example starts up the Windows Calculator.
// If the user minimizes the calculator and then
// tries to start it up again, the Calculator is
//brought to the top instead of starting up another instance of it.
//To use this in your titles, copy the following three lines into
//your frame as well as the watch for.  Then modify the watch
//for to watch for whatever button or event you have designated
//will launch the calculator.  You do not need to use the
//call program tool at the title design level for this to work.
|
WORD  wStatus;                               //Copy these 3 lines of
HWND  hwnd;                                  //code into your frame
char  szMsg[80];

Graphic File (calc.bmp) "OpenCalc"
Panel

//Copy this watch for into your frame and modify it to watch
//for whatever event you would like to launch the calculator
Watch for... "OpenCalc" LButtonClicked then...
```

Page 23 of textbook

Multimedia Authoring Metaphors

- 6. **Card/Scripting Metaphor**: Uses a simple index-card structure - easy route to producing applications that use hypertext or hypermedia; used in schools.


hyperstudio_example.stk
File Edit Move Tools Objects Colors Options Extras Help
Click here to start your game!
hyperstudio_example.stk
File Edit Move Tools Objects Colors Options Extras Help
Pick Your Character!
Frank
Bryanessa
James

Page 24 of textbook

Multimedia Authoring Metaphors

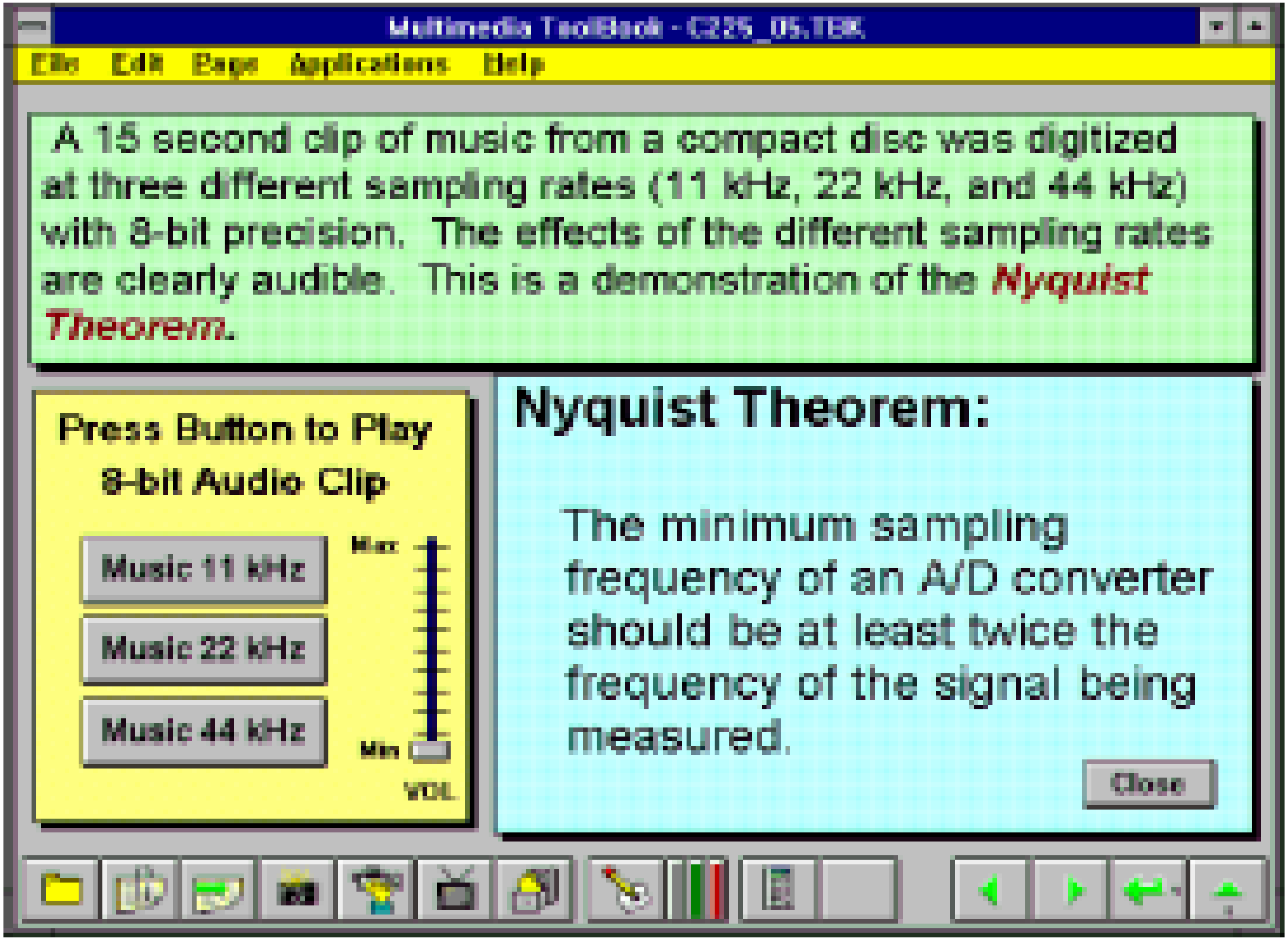
7. **Cast/Score/Scripting Metaphor**:

- Time is shown horizontally; like a spreadsheet: rows, or **tracks**, represent instantiations of characters in a multimedia production.
- Multimedia elements are drawn from a **cast** of characters, and **scripts** are basically event-procedures or procedures that are triggered by timer events.
- **Director**, by Macromedia, is the chief example of this metaphor. Director uses the **Lingo** scripting language, an object-oriented event-driven language.

Multimedia Presentation

- **Graphics Styles**: Human visual dynamics impact how presentations must be constructed.
  (a) **Color principles and guidelines**: Some color schemes and art styles are best combined with a certain theme or style. A general hint is to *not use too many colors*, as this can be distracting.
  (b) **Fonts**: For effective visual communication in a presentation, it is best to use large fonts (i.e., 18 to 36 points), and no more than 6 to 8 lines per screen (*fewer than on this screen!*).

Multimedia Presentation



Multimedia ToolBook - C225_05.TBK
File Edit Page Applications Help

A 15 second clip of music from a compact disc was digitized at three different sampling rates (11 kHz, 22 kHz, and 44 kHz) with 8-bit precision. The effects of the different sampling rates are clearly audible. This is a demonstration of the *Nyquist Theorem*.

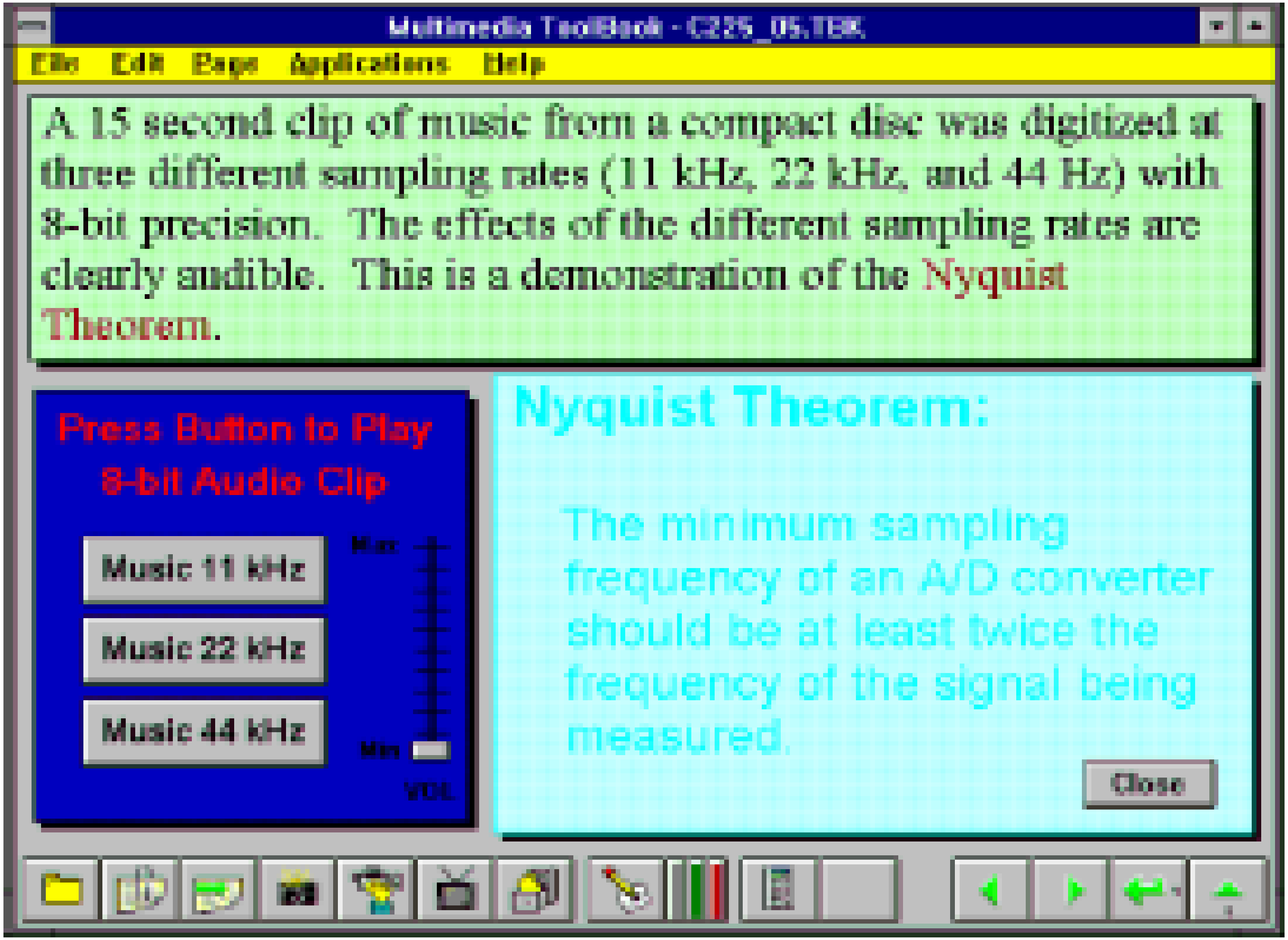
Press Button to Play 8-bit Audio Clip

Music 11 kHz
Music 22 kHz
Music 44 kHz
Max
Min
VOL

Nyquist Theorem:

The minimum sampling frequency of an A/D converter should be at least twice the frequency of the signal being measured.

Multimedia Presentation



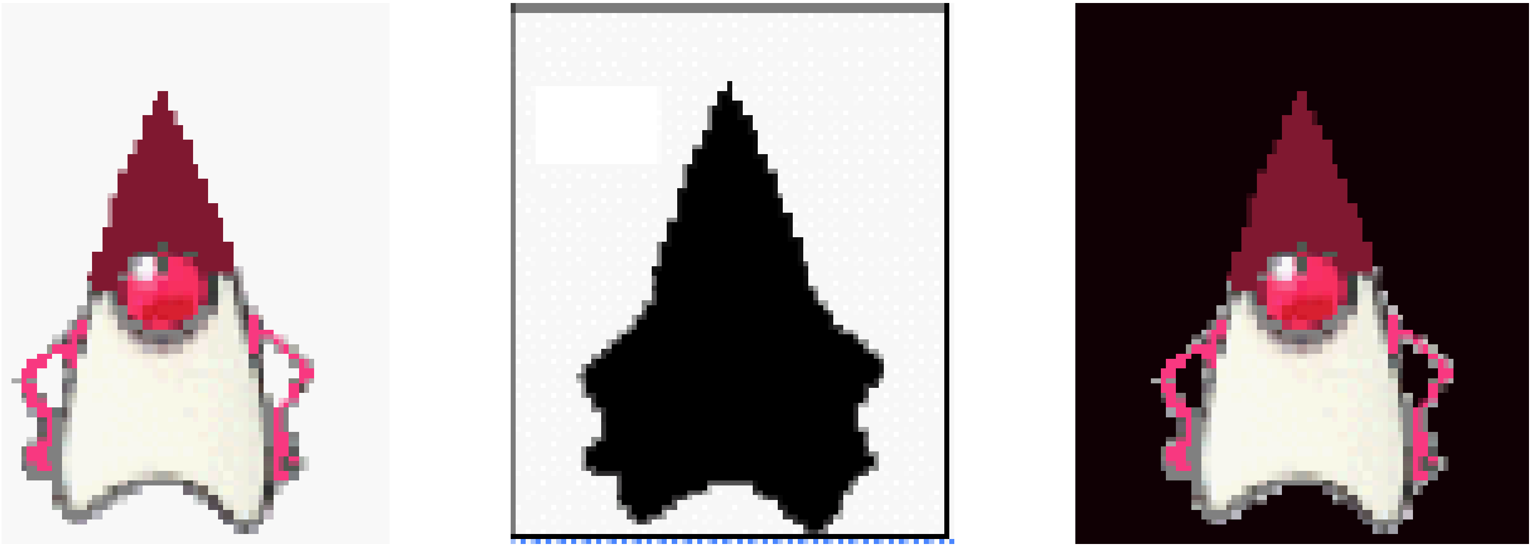
A 15 second clip of music from a compact disc was digitized at three different sampling rates (11 kHz, 22 kHz, and 44 Hz) with 8-bit precision. The effects of the different sampling rates are clearly audible. This is a demonstration of the Nyquist Theorem.

Nyquist Theorem:

The minimum sampling frequency of an A/D converter should be at least twice the frequency of the signal being measured.

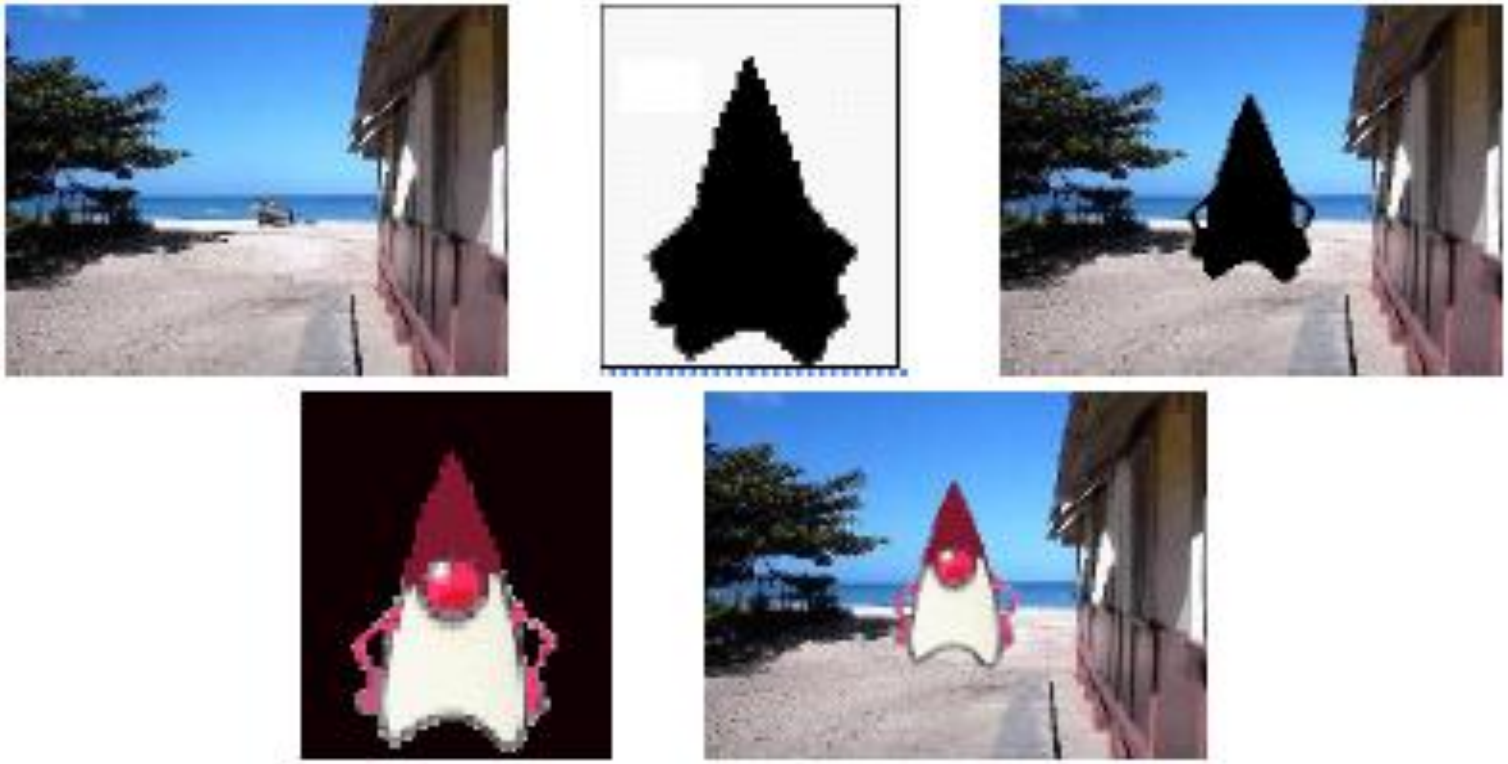
Sprite Animation

- **The basic idea**: We have an animation figure.
- Now create
  - – a 1-bit mask *M*, black on white,
  - – an accompanying *sprite S* with black background.



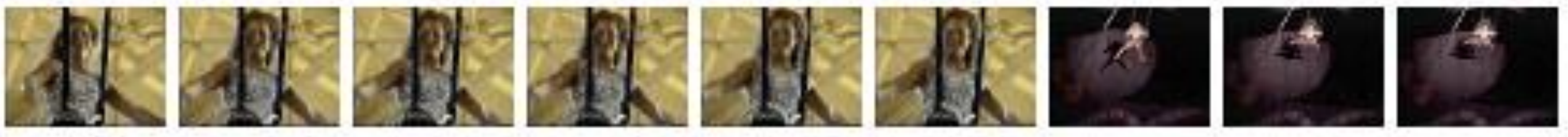
Sprite Animation

- We can overlay the sprite on a colored background *B*, by first ANDing *B* and *M*, and then ORing the result with *S.*



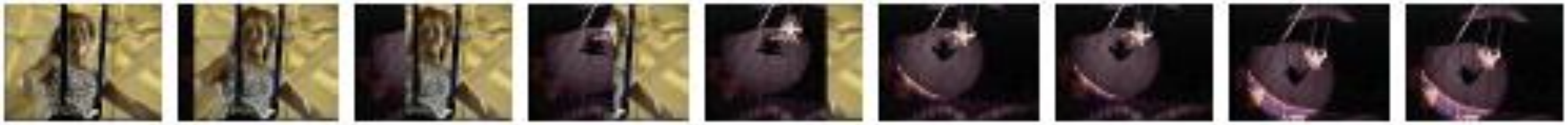
Video Transitions

- **Video transitions** signal “scene changes". There are many different types of transitions:
  1. **Cut**: an abrupt change of image contents formed by abutting two video frames consecutively. This is the simplest and most frequently used video transition.



Video Transitions

2. **Wipe**: a replacement of the pixels in a region of the viewport with those from another video. Wipes can be left-to-right, right-to-left, vertical, horizontal, like an iris opening, swept out like the hands of a clock, etc.



3. **Dissolve**: replaces every pixel with a mixture over time of the two videos, gradually replacing the first by the second. Most dissolves can be classified as two types:
cross dissolve and **dither dissolve**.

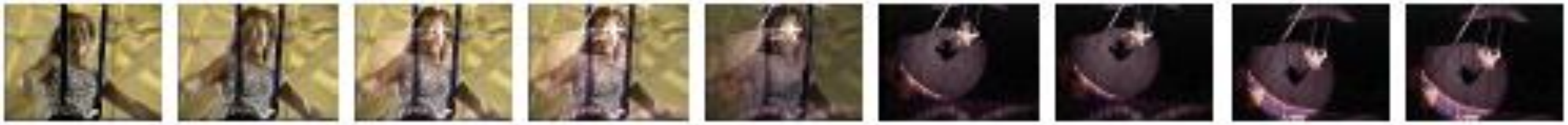
Type I: Cross Dissolve

Every pixel is affected gradually. It can be defined by:

$$\mathbf{D} = (1-\alpha(t)).\mathbf{A} \;+\; \alpha(t).\mathbf{B}$$

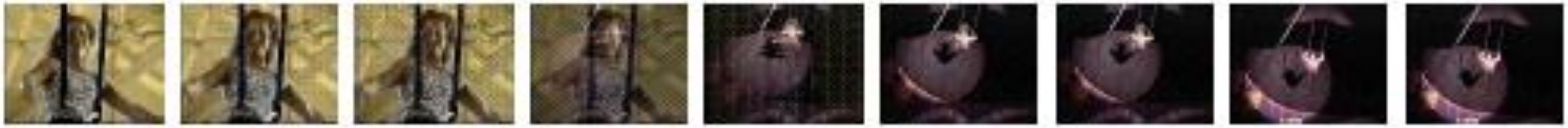
where **A** and **B** are the color 3-vectors for video A and video B. Here, α(*t*) is a transition function, which is often linear:

$$\alpha(t) = k.t \;\; \text{with } k.tmax = 1$$



Type II: Dither Dissolve

- Determined by (*t*), increasingly more and more pixels in video A will abruptly (instead of gradually as in Type I) change to video B.



Some Technical Design Issues

1. **Computer Platform**: Much software is ostensibly “portable“ but cross-platform software relies on run-time modules which may not work well across systems.
2. **Video format and resolution**: Some popular video formats - NTSC, PAL, and SECAM - are not compatible, so a conversion is required before a video can be played on a player supporting a different format.
3. **Memory and Disk Space Requirement**: At least 1 GB of RAM and 100 GB of hard-disk space should be available for acceptable performance and storage for multimedia programs.

Delivery Methods

- Not everyone has rewriteable DVD drives, yet.
- CD-ROMs: may be not enough storage to hold a multimedia presentation. As well, access time for CD-ROM drives is longer than for hard-disk drives.
- Electronic delivery is an option, but depends on network bandwidth at the user side (and at server). A streaming option may be available, depending on the presentation.

Automatic Authoring

- **Hypermedia documents**: Generally, three steps:

1. **Capture of media**: From text or using an audio digitizer or video frame-grabber; is highly developed and well automated.
2. **Authoring**: How best to structure the data in order to support multiple views of the available data, rather than a single, static view.
3. **Publication**: i.e. Presentation, is the objective of the multimedia tools we have been considering.

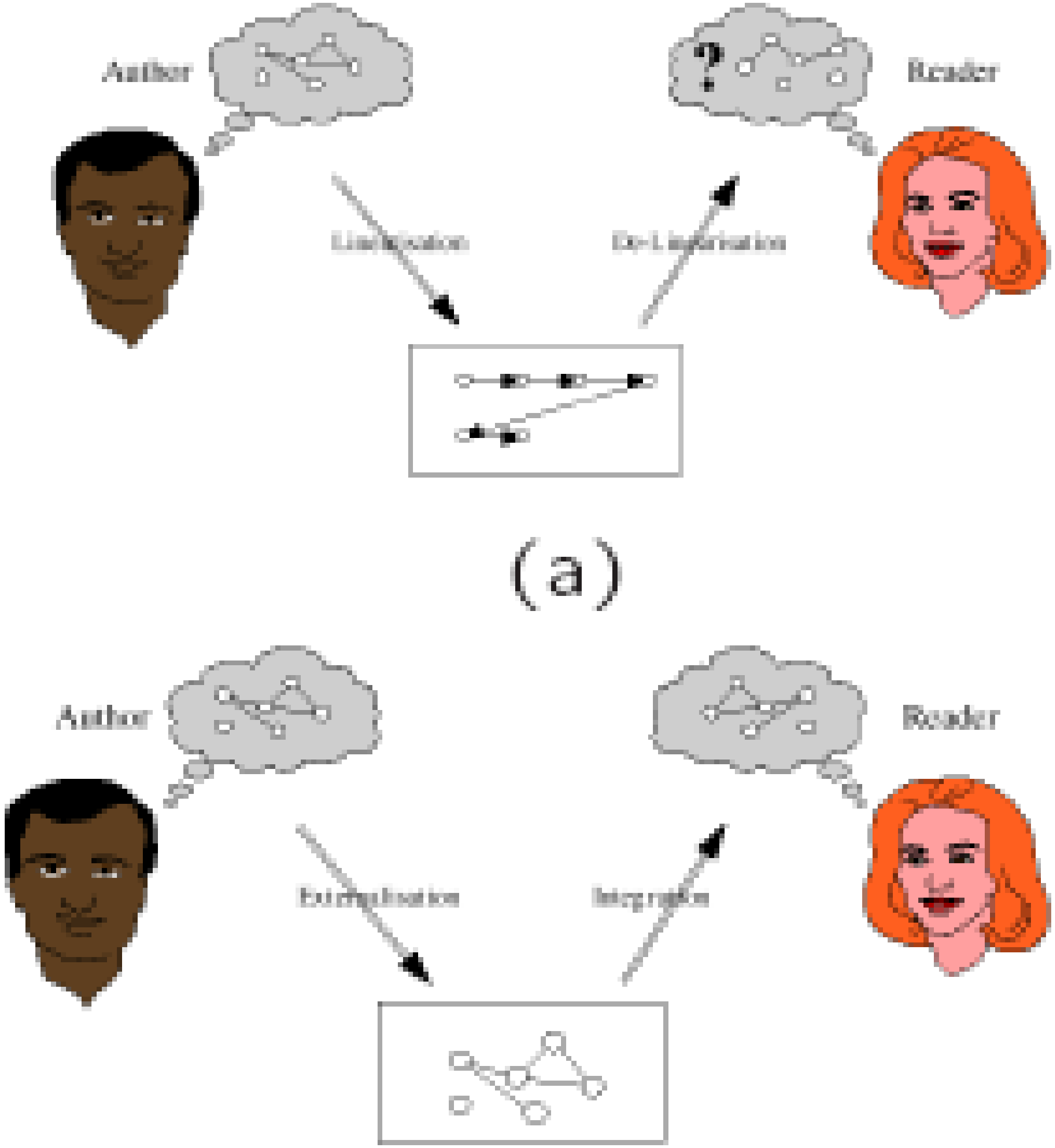
Externalization versus linearization

(a) Linearization represents the essential problem involved in communicating ideas
without using a hypermedia mechanism.

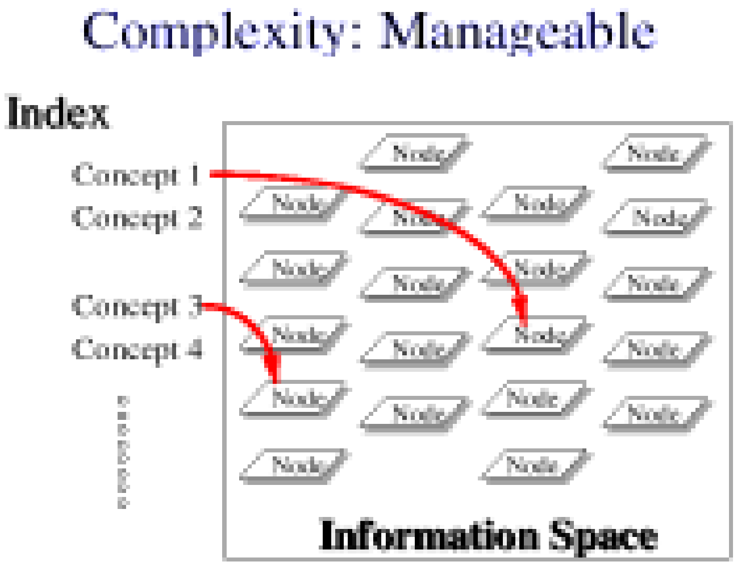
(b) In contrast, hyperlinks allow us the freedom to partially mimic the author's thought process (i.e., externalization).

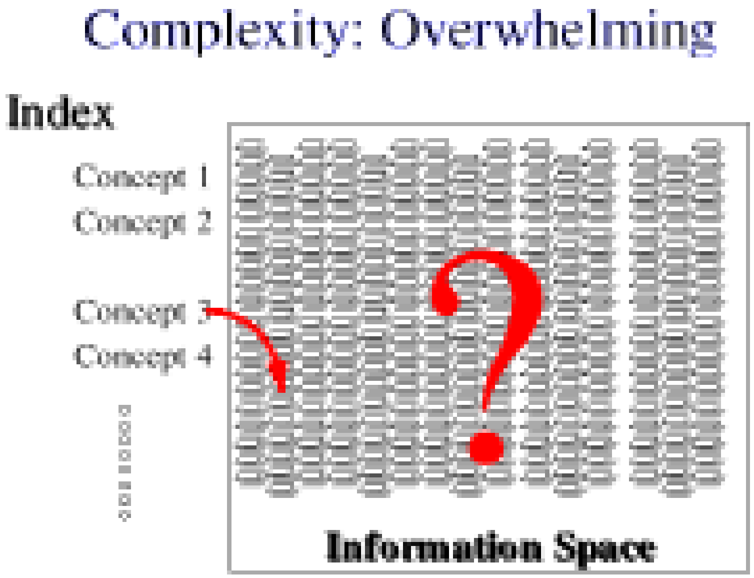
(c) Using, e.g., Microsoft Word, creates a hypertext version of a document by following the layout already set up in chapters, headings, and so on.
But problems arise when we actually need to extract **semantic** content automatically and find links and anchors (even considering just text and not images etc.)

Communication using hyperlinks



Complex information space

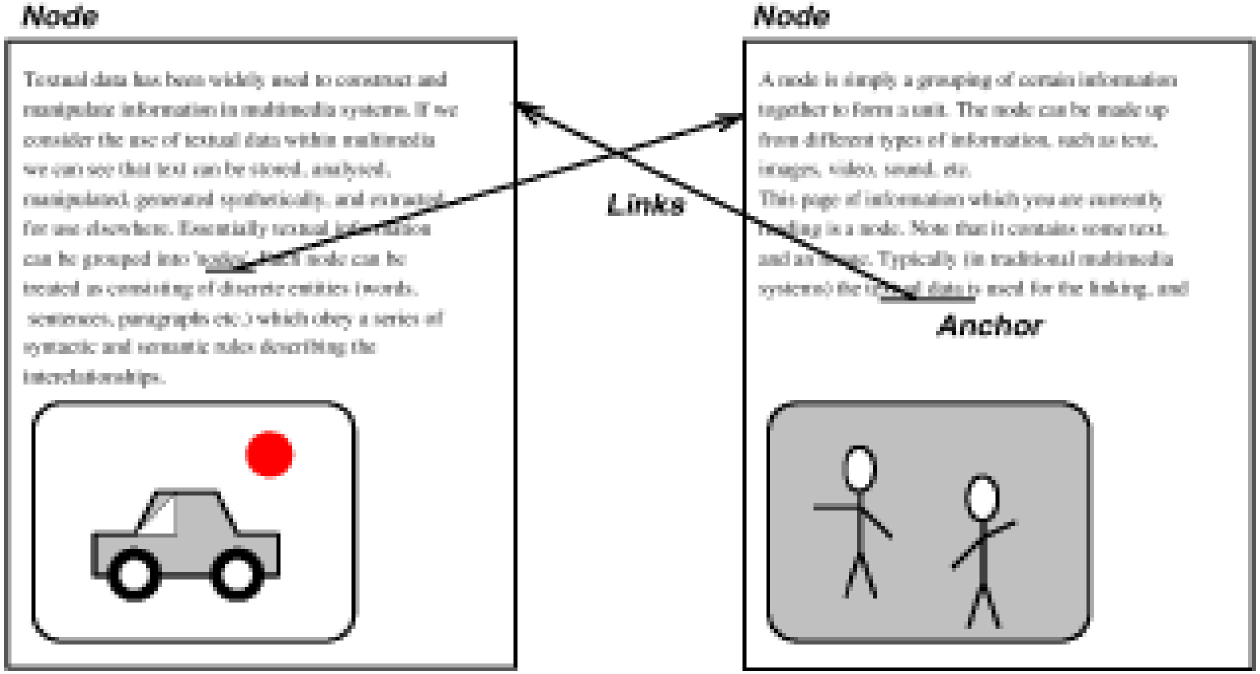




- Once a dataset becomes large we should employ database methods. The issues become focused on scalability (to a large dataset), maintainability, addition of material, and reusability.

Semi-automatic migration of hypertext

- The structure of hyperlinks for text information is simple: “nodes" represent semantic information and these are anchors for links to other pages.



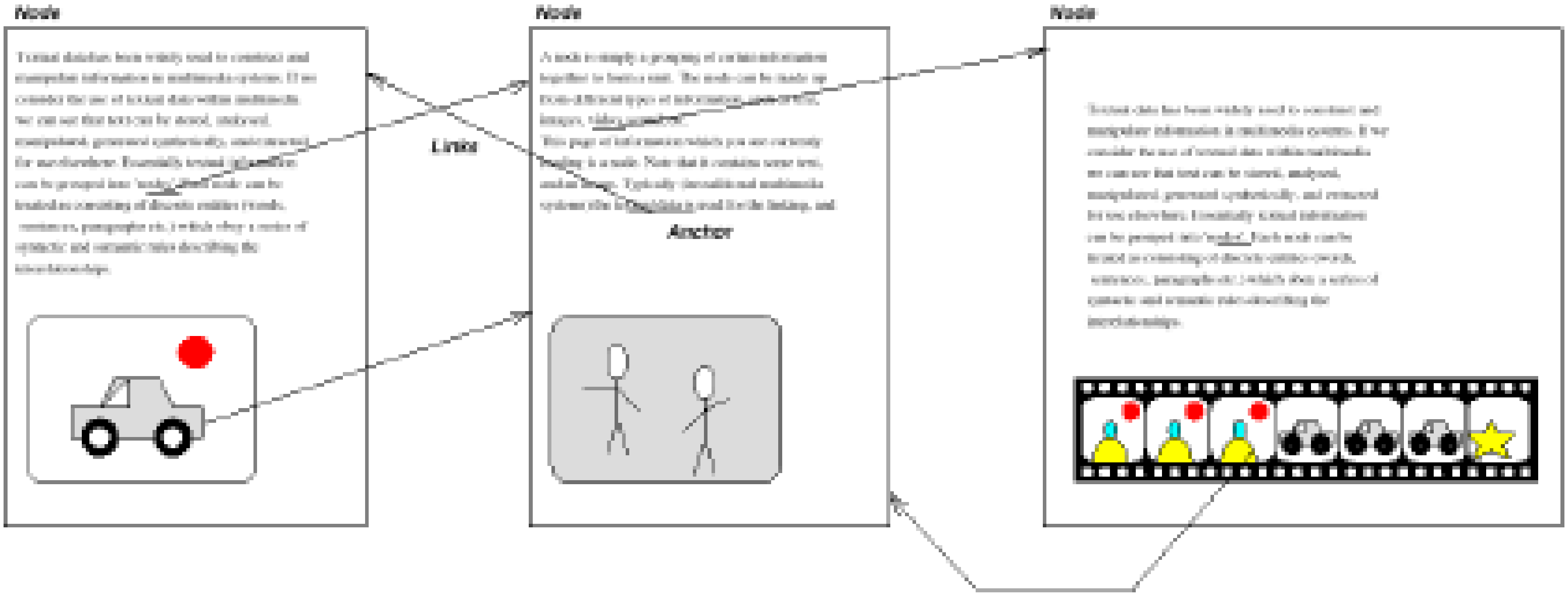
Textual data has been widely used to construct and manipulate information in multimedia systems. If we consider the use of textual data within multimedia we can see that text can be stored, analysed, manipulated, generated synthetically, and extracted for use elsewhere. Essentially textual information can be grouped into 'nodes'. Each node can be treated as consisting of discrete entities (words, sentences, paragraphs etc.) which obey a series of syntactic and semantic rules describing the interrelationships.

A node is simply a grouping of certain information together to form a unit. The node can be made up from different types of information, such as text, images, video, sound, etc.

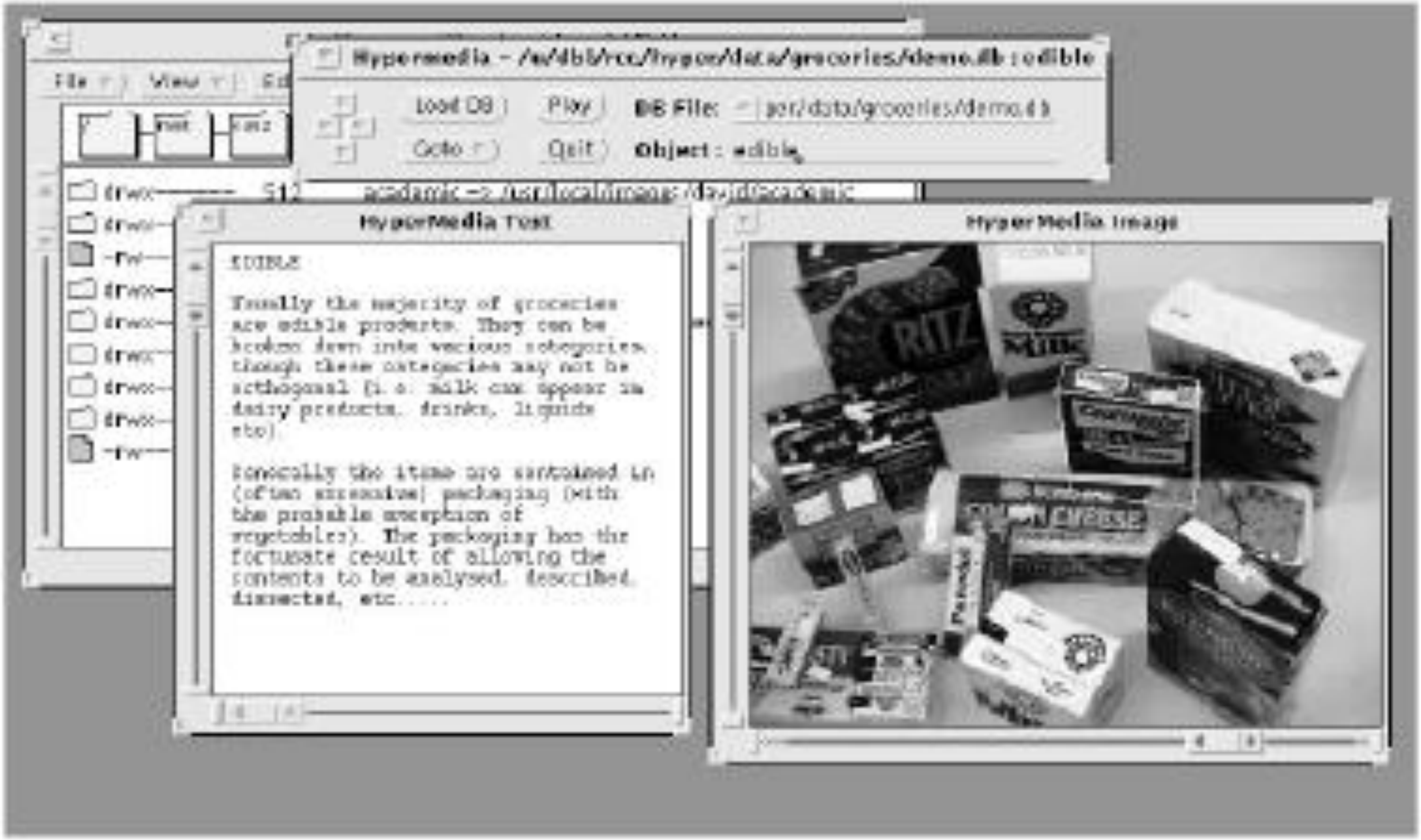
This page of information which you are currently reading is a node. Note that it contains some text, and an image. Typically (in traditional multimedia systems) the textual data is used for the linking, and

Hyperimages

- An automated method to help us produce true hypermedia



Can manually delineate syntactic image elements by masking image areas. Figure shows a “hyperimage", with image areas identified and automatically linked to other parts of a document.


Hypermedia – /u/dbl/rcc/hypon/data/groceries/demo.db : edible
Load DB
Play
Goto
Quit
File
View
HyperMedia Text
HyperMedia Image

DB File: per/data/groceries/demo.db

Object : edible

academic -> /usr/local/image/david/academic

EDIBLE

Usually the majority of groceries are edible products. They can be broken down into various categories, though these categories may not be orthogonal (i.e. milk can appear in dairy products, drinks, liquids etc).

Generally the items are contained in (often excessive) packaging (with the probable exception of vegetables). The packaging has the fortunate result of allowing the contents to be analysed, described, dissected, etc.....

Some Useful Editing and Authoring Tools

- One needs real vehicles for showing understanding principles of and creating multimedia. And straight programming in C++ or Java is not always the best way of showing your knowledge and creativity.
- Some popular authoring tools include the following:
  - Adobe Premiere 6
  - Macromedia Director 8 and MX
  - Flash 5 and MX
  - Dreamweaver MX
  - Cakewalk Pro Audio
- **Hands-on work in a Lab**

VRML
(Virtual Reality Modeling Language)

(a) **VRML**: conceived in the first international conference of the World Wide Web as a platform-independent language that would be viewed on the Internet.

(b) **Objective of VRML**: capability to put colored objects into a 3D environment.

(c) VRML is an interpreted language; however it has been very influential since it was the first method available for displaying a 3D world on the World Wide Web.

History of VRML

- VRML 1.0 was created in May of 1995, with a revision for clarification called VRML 1.0C in January of 1996:
  - VRML is based on a subset of the file inventor format created by Silicon Graphics Inc.
  - VRML 1.0 allowed for the creation of many simple 3D objects such as a cube and sphere as well as user-defined polygons. Materials and textures can be specified for objects to make the objects more realistic.

History of VRML

- The last major revision of VRML was VRML 2.0, standardized by ISO as VRML97:
  - This revision added the ability to create an interactive world. VRML 2.0, also called “Moving Worlds", allows for animation and sound in an interactive virtual world.
  - New objects were added to make the creation of virtual worlds easier.
  - Java and Javascript have been included in VRML to allow for interactive objects and user-defined actions.
  - VRML 2.0 was a large change from VRML 1.0 and they are not compatible with each other. However, conversion utilities are available to convert VRML 1.0 to VRML 2.0 automatically.

VRML Shapes

- VRML contains basic geometric shapes that can be combined to create more complex objects. Some of these shapes are:

- **Shape node** is a generic node for all objects in VRML.
- **Material node** species the surface properties of an object. It can control what color the object is by specifying the red, green and blue values of the object.

VRML Shapes

- There are three kinds of texture nodes that can be used to map textures onto any object:
  1. **ImageTexture**: The most common one that can take an external JPEG or PNG image file and map it onto the shape.
  2. **MovieTexture**: allows the mapping of a movie onto an object; can only use MPEG movies.
  3. **PixelTexture**: simply means creating an image to usewith ImageTexture within VRML.

VRML world

Openable-book VRML simple world

- The position of a viewpoint can be specified with the position node and it can be rotated from the default view with the orientation node.
- Also the camera's angle for its field of view can be changed from its default 0.78 radians, with the fieldOfView node.
- Changing the field of view can create a telephoto effect.

VRML world

- Three types of lighting can be used in a VRML world:
  - **DirectionalLight** node shines a light across the whole world in a certain direction.
  - **PointLight** shines a light from all directions from a certain point in space.
  - **SpotLight** shines a light in a certain direction from a point.
- The background of the VRML world can also be specified using the **Background** node.
- A **Panorama** node can map a texture to the sides of the world. A panorama is mapped onto a large cube surrounding the VRML world.

Animation and Interactions

- The only method of animation in VRML is by tweening - done by slowly changing an object that is specified in an interpolator node.
- This node will modify an object over time, based on the six types of interpolators: color, coordinate, normal, orientation, position, and scalar.
- (a) All interpolators have two nodes that must be specified: the **key** and **keyValue**.
- (b) The **key** consists of a list of two or more numbers starting with 0 and ending with 1, defines how far along the animation is.
- (c) Each key element must be complemented with a keyValue element: defines what values should change.

Animation and Interactions

- To time an animation, a **TimeSensor** node is used:
  - (a) **TimeSensor** has no physical form in the VRML world and just keeps time.
  - (b) To notify an interpolator of a time change, a ROUTE is needed to connect two nodes together.
  - (c) Most animation can be accomplished through the method of routing a TimeSensor to an interpolator node, and then the interpolator node to the object to be animated.
- Two categories of sensors can be used in VRML to obtain input from a user:
  - (a) **Environment sensors**: three kinds of environmental sensor nodes: VisibilitySensor, ProximitySensor, and Collision.
  - (b) **Pointing device sensors**: touch sensor and drag sensors.

VRML Specifics

(a) A VRML file is simply a text file with a “.wrl" extension.

(b) VRML97 needs to include the line #VRML V2.0 UTF8 in the first line of the VRML file - tells the VRML client what version of VRML to use.

(c) VRML nodes are case sensitive and are usually built in a hierarchical manner.

(d) All Nodes begin with “{" and end with “}" and most can contain nodes inside of nodes.

(e) Special nodes called group nodes can cluster together multiple nodes and use the keyword “children" followed by “[...]".

(f) Nodes can be named using DEF and be used again later by using the keyword USE. This allows for the creation of complex objects using many simple objects.

Simple VRML Example

A simple VRML example to create a box in VRML, accomplished by:

```
Shape {
  Geometry Box{}
}
```

- The Box defaults to a 2-meter long cube in the center of the screen. Putting it into a Transform node can move this box to a different part of the scene. We can also give the box a different color, such as red.

```
Transform { translation 0 10 0 children [
        Shape {
                Geometry Box{}
                appearance Appearance {
                        material Material {
                                diffuseColor 1 0 0
}}}]}
```