

CHƯƠNG TRÌNH KC 01
ĐỀ TÀI MÃ SỐ KC 01-14
-----&-----

ĐỀ TÀI THUỘC CHƯƠNG TRÌNH KHCN CẤP NHÀ NƯỚC KC 01
MÃ SỐ KC 01.14

**NGHIÊN CỨU PHÁT TRIỂN ỨNG DỤNG
CÔNG NGHỆ ĐA PHƯƠNG TIỆN**

Chủ nhiệm đề tài: PGS.TS. Nguyễn Cát Hồ

BÁO CÁO NHÁNH:

**“ QUY TRÌNH XÂY DỰNG PHẦN MỀM
MÔ PHỎNG THÍ NGHIỆM ẢO ĐA PHƯƠNG TIỆN”**

Chủ trì: PGS. TS. Nguyễn Đình Hóa

6352-9
20/4/2007

HÀ NỘI, 4/2005

ĐỀ TÀI KC. 01-14
Viện Công nghệ Thông tin, Đại học Quốc gia Hà Nội

TÀI LIỆU:

QUY TRÌNH XÂY DỰNG PHẦN MỀM
MÔ PHỎNG THÍ NGHIỆM ẢO ĐA PHƯƠNG TIỆN

Người viết: Lê Việt Hà

HÀ NỘI, 4-2005

Mục lục

1. Mở đầu	4
2. Mục đích	6
3. Đối tượng sử dụng	8
4. Sản phẩm đa phương tiện hỗ trợ dạy học trên máy tính cá nhân	8
5. Quy trình chung xây dựng sản phẩm đa phương tiện	8
5.1. Pha chuẩn bị sản xuất	9
5.1.1 Đề xuất sản phẩm thực hiện	10
5.1.2 Thiết kế khung sản phẩm	11
5.1.3 Thu thập, tìm kiếm dữ liệu	12
5.1.4 Yêu cầu tài nguyên	14
5.2. Pha sản xuất	15
5.3. Pha sau sản xuất	17
6. Kiến trúc nền cho sản phẩm đa phương tiện	18
6.1. Nguyên tắc chung xây dựng sản phẩm đa phương tiện	18
6.2. Nguyên tắc xây dựng nội dung sản phẩm đa phương tiện	19
6.3. Thiết kế giao diện tương tác người dùng máy tính	24
7. Đánh giá hiện trạng và lựa chọn kỹ thuật	25
8. Lập kế hoạch	29
9. Nội dung thực hiện	30
9.1. Cấu trúc sản phẩm	30
9.2. Viết kịch bản	30
9.3. Cài đặt phần mềm	Error! Bookmark not defined.
9.4. Yêu cầu phần cứng	Error! Bookmark not defined.
9.5. Xây dựng kho dữ liệu đồ họa cơ bản	Error! Bookmark not defined.
9.5.1 Xây dựng dữ liệu văn bản	Error! Bookmark not defined.
9.5.2 Xây dựng dữ liệu ảnh	Error! Bookmark not defined.
9.5.3 Xây dựng dữ liệu hoạt hình	Error! Bookmark not defined.
9.5.4 Xây dựng dữ liệu âm thanh	Error! Bookmark not defined.
9.5.5 Xây dựng dữ liệu video	Error! Bookmark not defined.
9.6. Tích hợp dữ liệu	Error! Bookmark not defined.

9.7.	Kiểm tra	Error! Bookmark not defined.
10.	Kết luận	Error! Bookmark not defined.
11.	Phụ lục - Xây dựng phần mềm Hiện tượng khúc xạ ánh sáng	Error! Bookmark not defined.
11.1.	Kế hoạch xây dựng phần mềm	Error! Bookmark not defined.
11.2.	Nội dung thực hiện	Error! Bookmark not defined.
11.2.1	Thống nhất kịch bản	Error! Bookmark not defined.
11.2.2	Cài đặt phần mềm	Error! Bookmark not defined.
11.2.3	Yêu cầu phần cứng	Error! Bookmark not defined.
11.2.4	Xây dựng và quản lý kho dữ liệu đồ họa cơ bản	Error! Bookmark not defined.
11.2.5	Lập trình	Error! Bookmark not defined.
11.2.6	Tích hợp dữ liệu	Error! Bookmark not defined.
11.2.7	Chương trình	Error! Bookmark not defined.
11.2.8	Kiểm tra	Error! Bookmark not defined.
11.2.9	In lên CD-ROM	Error! Bookmark not defined.
11.3.	Kết quả	Error! Bookmark not defined.

Quy trình xây dựng phần mềm mô phỏng thí nghiệm ảo đa phương tiện

1. Mở đầu

Thông tin khoa học kĩ thuật đóng một vai trò vô cùng quan trọng trong quá trình dạy học. Tin học tương tác với giáo dục thỏa mãn nhu cầu giúp cho nhiều người nắm bắt được tri thức phản ánh những thành tựu cơ bản của nền văn minh, tạo nên cơ sở giáo dục dưới mọi hình thức. Sẽ không hiểu lĩnh vực giáo dục một cách cặn kẽ nếu như không tính đến mối liên quan của nó đối với tin học. Ngược lại trong tổng thể các vấn đề xã hội và nhân văn, tin học nghiên cứu một khía cạnh đặc biệt liên quan đến sự phát triển của lĩnh vực giáo dục, đến tình hình hiện thời và những xu hướng mới nhất của nó. Trong lĩnh vực giáo dục cũng như các lĩnh vực quan trọng khác của đời sống xã hội, tin học và các phương tiện máy tính ngày càng đóng vai trò to lớn.

Trong khi xây dựng phần mềm mô phỏng thí nghiệm ảo đa phương tiện chúng ta cần quan tâm đến khái niệm hệ thống đa phương tiện số. Thông tin được truyền đạt bằng các hệ thống truyền đa phương tiện, gọi tắt là thông tin đa phương tiện có thể thể hiện ở những dạng sau:

- Văn bản (text).
- Đồ hoạ (graphic).
- Hoạt hình (animation).
- Hình ảnh (image).
- Đoạn phim (Video).
- Âm thanh (Audio).

Thuật ngữ Media được hiểu là sự hiển thị, lưu trữ và truyền dữ liệu. Nó dùng để chỉ các thực thể như là audio, video, điện thoại, tạp chí - chúng là những thực thể không chỉ mang thông tin đơn thuần mà là những hệ thống tương đối phức tạp, có cơ cấu, có đối tượng cụ thể. Dạng khác của Media được nói đến đó là các bức tranh, tác phẩm điêu khắc và các đối tượng khác mà có khả năng mô tả và truyền đạt thông tin. Loại truyền thông trực tiếp, từ miệng người này đến tai người kia, không sử dụng thành phần (media) trung gian. Không khí truyền các chấn động âm thanh không phải là một media mà nó chỉ mang tính chất vật lý thông thường đó là làm công việc tải thông tin.

Media có thể được phân thành 2 nhóm lớn, đó là:

- Sự biểu hiện : đó là sự trình bày thông tin dưới dạng có thể hiểu được. Chúng bao gồm các kí tự, từ ngữ, hình ảnh, bảng biểu, ma trận, lưu đồ và các đối tượng có tính chất tương tự.
- Truyền thông: được hiểu là sự truyền thông tin từ một nguồn xa và hầu như không có thời gian trễ. Nhóm này bao gồm các thiết bị truyền thông như đường điện thoại, đài, tivi,... và các thiết bị lưu giữ thông tin như tạp chí, băng từ, đĩa từ, v.v...

Tuy nhiên trong hầu hết các trường hợp thì sự biểu hiện và sự truyền là không thể tách rời. Thông tin hoặc sự mô tả thông tin được truyền đến người nhận bằng các thiết bị truyền tin.

Đa phương tiện là kĩ thuật mô phỏng và sử dụng đồng thời nhiều dạng phương tiện chuyển hóa thông tin và các tác phẩm từ các kĩ thuật đó. Nói cách khác thì đa phương tiện là sự phối hợp của văn bản, hình ảnh, âm thanh; có nghĩa là sự truyền thông đồng bộ nhiều dạng dữ liệu và cho phép người tham gia truyền thông tương tác với chúng.

Liên quan đến định nghĩa đa phương tiện cần lưu ý những khía cạnh sau:

- Thông tin cần phải được số hoá: đây là điểm quan trọng nhất, các kiểu thông tin mà chúng ta nhận vào như văn bản, âm thanh, hình ảnh... thường tồn tại dưới dạng tín hiệu tương tự. Để sử dụng các thông tin tương tự này trên máy tính thì đầu tiên chúng ta phải chuyển đổi nó về dạng tín hiệu số.
- Sử dụng mạng máy tính: các thiết bị đầu cuối đa phương tiện được sử dụng ở nhà hoặc tại văn phòng và các tổ chức xã hội phải được kết nối với nhau bởi mạng để có thể trao đổi thông tin với nhau. Đó là yếu tố cần thiết để phát triển một xã hội đa phương tiện.
- Sử dụng phần mềm có tương tác: phần mềm sáng tác sản phẩm đa phương tiện phải phối hợp, tương tác với các phần mềm khác. Do đó dữ liệu sử dụng sẽ phong phú hơn. Đồng thời nó có thể kết hợp được với các ngôn ngữ lập trình bậc cao để đưa ra sản phẩm ưu việt hơn.
- Phải thiết kế giao diện người-máy: hiện thực về một xã hội thông tin sẽ không xuất hiện trừ khi mọi người có thể tác động tới thông tin bằng những thao tác đơn giản. Chẳng hạn, người sử dụng có thể tác động đến chương trình một cách dễ dàng thông qua bàn phím, chuột... Để đạt được điều này, công nghệ cần phải tạo cho nó khả năng là bất kì ai cũng có thể tác động tới thiết bị đầu cuối.

Trong định nghĩa đa phương tiện nói đến việc sử dụng đồng thời các dạng dữ liệu là chúng ta muốn nói đến yếu tố thời gian. Có nghĩa là ở cùng một thời điểm các dữ liệu cùng được khai thác sử dụng. Chính điều này khiến người ta phải nói tới khái niệm

đồng bộ. Khái niệm đồng bộ là một trong những khái niệm rất quan trọng, có tính cốt lõi của đa phương tiện. Đồng bộ là sự sắp đặt các thay đổi hay biến đổi của đối tượng (còn gọi là sự kiện) theo trật tự thời gian sao cho ở cùng một thời điểm, các sự kiện có cùng trật tự thời gian cùng được xuất hiện.

Khi nói đến giáo dục thì người ta nghĩ ngay đến sự truyền đạt tri thức, chỉ thế thì chưa đủ, tri thức đó cần phải chuyển thành động lực thúc đẩy con người hành động, hình thành nên một loạt những hành vi sẽ diễn ra trong mọi điều kiện có thể.

Thiết kế các phần mềm mô phỏng thí nghiệm ảo đa phương tiện đã được nhiều cơ sở trong nước và trên thế giới thực hiện. Mặc dù triển vọng và khả năng ứng dụng là rất lớn nhưng chưa hoàn toàn hấp dẫn được người sử dụng như mong muốn.

Vì vậy việc tìm cách nâng sự hấp dẫn, mức độ thuận tiện cũng như những ưu thế tương tác người máy còn là một thách thức kỹ thuật tạo phần mềm loại này. Tăng cường việc tích hợp tối đa các khả năng có thể của kỹ thuật Multimedia trong cùng một phần mềm là ý tưởng đang được nhiều phòng thí nghiệm hướng tới. Trong phần này chúng tôi trình bày quy trình xây dựng sản phẩm đa phương tiện và áp dụng vào việc xây dựng các thí nghiệm ảo phục vụ giảng dạy của giáo viên phổ thông. Trong đó sẽ đi chi tiết quy trình tích hợp phương tiện nghe nhìn kỹ thuật số với các kỹ thuật mô phỏng trên máy tính, kỹ thuật văn bản, kỹ thuật tương tác... để tạo nên phần mềm dạy học hoặc phổ biến kiến thức. Các giải pháp này được dùng trong giáo dục và được định hướng phát triển sao cho thật đơn giản ứng dụng được cho việc phổ cập tri thức tới học sinh.

2. Mục đích

Việc xây dựng phần mềm mô phỏng thí nghiệm ảo đa phương tiện có tính tương tác cao nhằm phục vụ việc dạy và học với tính năng sư phạm phù hợp.

Ngoài ra chúng tôi có tiến hành xây dựng kho dữ liệu các yếu tố đồ họa số hoá cơ bản (tĩnh và video) phục vụ cho việc phát triển các thí nghiệm mô phỏng sau này.

Phần mềm mô phỏng thí nghiệm ảo đa phương tiện có các chỉ tiêu và yêu cầu kỹ thuật cần đạt tới như sau:

a. Chức năng:

- Chức năng hoạt động
- Quản trị các thí nghiệm ảo
- Thực hiện các thí nghiệm ảo trong giảng dạy, thực hành và làm bài tập thí nghiệm

Yêu cầu kỹ thuật:

- Bảo đảm tính pháp lý sư phạm
 - Tính chân thật, đúng đắn, gần hiện thực
 - Có tính tương tác cao phù hợp với yêu cầu giáo viên và học sinh thực hành
 - Tính mỹ thuật
- b. Giao diện**
- Thân thiện, dễ sử dụng
 - Tương tác thông qua menu
 - Làm nổi bật tính trực quan, tính đa dạng trong thể hiện nội dung
- c. An toàn và bảo mật**
- Sản phẩm phần mềm được mã hoá dưới dạng mã máy.
 - Có biện pháp chứng thực bản quyền
 - Có cơ chế bảo vệ CSDL
- d. Tính chuẩn hoá**
- Các dữ liệu Multimedia (video, hình ảnh..) được chuyển thành các dạng chuẩn quốc tế và chuẩn công nghiệp.
 - Chuẩn hoá trong thiết kế giao diện với người dùng.
 - Thống nhất sử dụng công cụ MySQL mã nguồn mở.
- e. Tính mở, tính khả chuyển**
- Thiết kế chương trình có tính module
 - Bảo đảm tính khái quát cho từng lớp các bài thí nghiệm tương tự
 - MySQL mã nguồn mở.
- f. Môi trường và công cụ cài đặt**
- Chương trình được cài đặt trên hệ điều hành phổ dụng Windows NT, Windows 9X và 2000.
- g. Tài liệu kỹ thuật đi kèm**
- Tài liệu mô tả nội dung
 - Tài liệu hướng dẫn sử dụng

3. Đối tượng sử dụng

Sản phẩm được xây dựng chủ yếu nhằm phục vụ hỗ trợ giáo dục đào tạo ở bậc phổ thông. Đối tượng sử dụng phần mềm chính là giáo viên và học sinh. Họ chính là những người trực tiếp tham gia thí nghiệm, tương tác với các đối tượng trong môi trường hiện tượng.

4. Sản phẩm đa phương tiện hỗ trợ dạy học trên máy tính cá nhân

Sản phẩm công nghệ đa phương tiện đã và đang xâm nhập ngày càng sâu rộng vào mọi lĩnh vực của đời sống xã hội. Có thể nói sản phẩm của công nghệ này có mặt ở khắp mọi nơi, từ công sở đến gia đình. Nó xuất hiện trong nhiều lĩnh vực từ giáo dục, y tế, vui chơi giải trí đến nghiên cứu khoa học...

Mô phỏng thí nghiệm bằng đa phương tiện tăng cường khả năng quan sát nhận thức của người học. Khi làm việc kiểu tương tác, người học vẫn thường xuyên đọc các văn bản trên màn hình để có thể chuyển dịch từ giai đoạn này sang giai đoạn khác, đồng thời vẫn thường xuyên đưa ra can thiệp bằng văn bản vào máy. Nhờ đó mà cách học tương tác, đào tạo bằng đa phương tiện ngày càng phổ biến.

- Việc thiết kế và sử dụng xây dựng phần mềm mô phỏng thí nghiệm ảo đa phương tiện có thể:
 1. Xây dựng thí nghiệm ảo theo kiểu lắp ráp các khối đã có sẵn.
 2. Chất lượng tiếp thu bài thí nghiệm sẽ cao hơn vì có nhiều hình ảnh minh họa mà ở phương pháp giảng dạy truyền thống có rất ít.
 3. Người học có thể học đi học lại một bài thí nghiệm nào đó nếu chưa hiểu hoặc có thể lướt qua những thí nghiệm mà họ cho là dễ chứ không nhất thiết phải ngồi học như đối với giờ giảng của giáo viên.
 4. Tìm kiếm thông tin một cách nhanh chóng nhờ các mối liên kết trong mỗi thí nghiệm.
 5. Người học có thể học vào thời điểm nào là tùy thích chứ không bị phụ thuộc vào một thời khoá biểu định sẵn.
 6. Mỗi bài thực hành thí nghiệm có thể kết thúc nhanh hay chậm tùy thuộc vào khả năng của người dùng chứ không bị hạn chế như bài học ở trường.

5. Quy trình chung xây dựng sản phẩm đa phương tiện

Mục đích của việc tạo sản phẩm đa phương tiện trên máy tính là tạo sự giao tiếp với văn bản. Sự chuyển động, hình ảnh, âm thanh, âm nhạc và người dùng có thể lựa chọn một

cách độc lập, riêng biệt thông quan các loại dữ liệu này. Cách trình bày này hầu hết đều thể hiện được đặc trưng, vẻ đẹp độc đáo của vấn đề. Nếu chất lượng trên không được hiện hữu rõ ràng, thì khó ai có thể nhận được các thông điệp được truyền tải trong đó.

Sáng tác sản phẩm đa phương tiện là một công việc khá lý thú và cũng đòi hỏi nhiều chuyên môn phối hợp. Quá trình sáng tác bao gồm 3 bước chính: chuẩn bị sản xuất, sản xuất, sau sản xuất.

Ba phần này bao gồm toàn bộ các phần trong quá trình tạo sản phẩm, và mỗi phần đều có các ưu thế quản lý khác nhau. Để trở thành một công cụ quản lý, những phần trong quy trình tạo sản phẩm cần tạo khung phản chiếu sản phẩm và được thảo luận/đánh giá trước và sau khi hoàn thành. Tuy nhiên chúng cũng được biến đổi trong mỗi sản phẩm và đội phát triển cho phù hợp, do vậy những mô tả dưới đây không phải luôn luôn tuyệt đối.

Các quá trình tạo sản phẩm cần thực hiện liên tục. Trước bất kỳ dãy thiết kế hoặc công việc nào được thực hiện, mọi người có liên quan trong quá trình xây dựng sản phẩm cần hiểu rõ vấn đề đó được làm như thế nào và tại sao phải làm như vậy. Hiểu nhầm hoặc không đồng ý về quá trình thực hiện dự án có thể làm chậm tiến trình.

5.1. Pha chuẩn bị sản xuất

Đây là pha đầu tiên của quy trình sản xuất sản phẩm đa phương tiện. Trong bước này ta phải xác định được phạm trù công việc và đối tượng sử dụng sản phẩm này. Xác định phạm trù công việc tức là chọn tên của sản phẩm. Tùy theo tính chất mà sản phẩm đó sẽ mang tên với ý nghĩa: tên sản phẩm, tên mang thông tin, tên sáng tác hoặc tên giáo dục. Sau đó dựa theo tên của sản phẩm mà ta xác định nội dung cơ bản của nó. Việc cuối cùng trong pha chuẩn bị sản xuất là xây dựng nhóm thực hiện. Trong nhóm thực hiện trước hết phải có một chủ nhiệm và một đạo diễn, thành phần còn lại trong nhóm có thể là người mới hoặc người cũ từ nhóm khác.

- Chọn tên:
 - Tên sản phẩm
 - Tên mang thông tin
 - Tên giải trí
 - Tên sáng tác
 - Tên giáo dục
- Xác định người dùng: quyết định thành công và gợi ý sáng tác cho sản phẩm sẽ phát triển

- Xác định nội dung theo tên sản phẩm
- Xây dựng nhóm thực hiện:
 - Chủ nhiệm
 - Đạo diễn
 - Người mới
 - Người cũ từ nhóm khác

Trước tiên chúng ta cần có khái niệm cơ bản về quy trình, tiềm năng của sản phẩm đa phương tiện. Quy trình này được xây dựng được coi như công cụ xây dựng các sản phẩm mô phỏng đa phương tiện sau đó.

- Giao tiếp thông tin qua đa phương tiện

Bước đầu tiên phải sắp xếp tạo kế hoạch những gì ta cần làm và kế hoạch làm như thế nào. Cần chú ý tới:

- Trình bày thông tin
 - Không chỉ là việc trình diễn thông tin mà cần phải thể hiện mối quan hệ về mặt nội dung giữa chúng
 - Thường sử dụng phương pháp nhấn mạnh bằng hình ảnh hơn là bằng lời.
- Cấu trúc
 - Phần mềm được xây dựng cần có cấu trúc rõ ràng
 - Nội dung thông tin được tạo cần theo giác quan con người
 - Cấu trúc xây dựng sao cho có thể sử dụng lại
- Nguyên tắc phân loại dựa trên kinh nghiệm và nhận thức đúng: theo đó giai đoạn chuẩn bị tạo một sản phẩm đa phương tiện lần lượt theo các bước sau: kiến thức, nhận thức đầy đủ về vấn đề cần làm, ứng dụng, phân tích, tổng hợp, đánh giá

5.1.1 Đề xuất sản phẩm thực hiện

Đề xuất kế hoạch là bước khởi đầu khi xây dựng sản phẩm. Việc phát triển một đề xuất liên quan tới xác định vùng chương trình giảng dạy. Mục đích của việc đề xuất kế hoạch thực hiện là cho phép chuyên gia trong lĩnh vực nào đó có thể truyền đạt kiến thức

của họ tới người học thông qua sản phẩm đa phương tiện với tài liệu dự định sẽ hoàn thành như thế nào và học sinh sẽ tương tác với tài liệu đó ra sao.

Kế hoạch đề án cần đặt ra các vấn đề sau:

- Sản phẩm cuối cùng sẽ phản ánh quá trình tạo ra nó
- Quá trình viết đề án sẽ mô tả:
 - Đề án ban đầu
 - Đích/ mục tiêu
 - Khán giả
 - Khái niệm, chủ đề
 - Phương tiện phân phát
 - Môi trường sản xuất
 - Ngân sách
 - Lịch sản xuất

Trong đề án ban đầu phải trả lời được về tính ưu việt khi sử dụng đa phương tiện để phát triển sản phẩm, bởi vì có rất nhiều lựa chọn trong đó đa phương tiện chỉ là một lựa chọn. Tiếp đến là phải xác định sản phẩm đa phương tiện là sản phẩm công cộng hay dùng riêng và cần đáp ứng yêu cầu về dạng sản phẩm. Có hai lý do chính để chúng ta đề xuất kế hoạch thực hiện đề án đa phương tiện là đưa ra các yêu cầu cần đạt được và thể hiện đề án rõ ràng bằng trực quan nội dung cần truyền tải.

Lưu ý: Trong trường hợp này giáo viên và học sinh là khía cạnh quan trọng cần chú ý vì họ chính là đối tượng mà đa phương tiện nhằm vào. Phải luôn luôn chú ý vì yếu tố này là động, thay đổi. Khi sử dụng các dữ liệu điều tra nên dùng dữ liệu hiện tại, không nên dựa trên dữ liệu điều tra trong quá khứ

5.1.2 Thiết kế khung sản phẩm

Thiết kế khung sản phẩm cung cấp nhiều điểm chính trong quá trình xây dựng sản phẩm. Chúng tôi đưa ra một số nguyên tắc định hướng chung khi thiết kế khung sản phẩm là:

1. Quyết định mục tiêu sản phẩm cần đạt: chúng ta muốn học sinh học được gì qua sản phẩm đa phương tiện? Chúng có thể thao tác được những phần gì? Một phân tích cần thiết phải có các yêu cầu này.

2. Phân tích mục tiêu truyền đạt kiến thức: sau khi phân tích các mục tiêu cần đạt, cần phân loại chúng. Thao tác từng bước của người dùng sẽ thể hiện rõ các mục tiêu này. Chúng ta có thể tạo một sơ đồ mô tả những vấn đề đó và mối quan hệ giữa chúng.
3. Phân tích người học và phạm vi nội dung thể hiện: nhà sản xuất cần biết người học đã biết những gì, ngữ cảnh họ xây dựng có phù hợp với tâm lý, trình độ, văn hóa của người học không?
4. Thể hiện mục đích khách quan: viết câu lệnh đặc biệt mô tả người học có thể làm được những gì theo hướng dẫn. Đặc tả tiêu chuẩn thể hiện thành công cho mỗi kỹ năng
5. Phát triển phương tiện đánh giá: sử dụng trình diễn khách quan, kỹ thuật này đánh giá khả năng thao tác của người học.

5.1.3 Thu thập, tìm kiếm dữ liệu

Một cách thể hiện đa phương tiện là liệt kê các dữ liệu thu thập được. Một số thành phần cơ bản trong một thể hiện đa phương tiện là: văn bản(text), hình ảnh(image), âm thanh(sound), hình động(animation) và phim(movie)

a. Dữ liệu văn bản

Chúng ta có thể tạo văn bản bằng các cách khác nhau. Cách đơn giản nhất là ta tạo trực tiếp bằng công cụ soạn thảo văn bản trong phần mềm trộn (tạo file video), hoặc là sử dụng file văn bản mở rộng. Định dạng của file văn bản hay được dùng: DOC, TXT,...

b. Dữ liệu ảnh

Ảnh tĩnh là hình ảnh được số hoá hay ảnh chụp (bằng máy quét hay máy ảnh số). Muốn lên kế hoạch về ảnh tĩnh đòi hỏi tổ chức nêu lên các nhu cầu về ảnh...Các chi tiết trong danh sách ảnh tĩnh là: số ảnh, tên file, số cảnh, kích thước file, màu sắc, ghi chú.

Ảnh trong thực tế là một ảnh liên tục về không gian và giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính cần thiết phải tiến hành số hoá ảnh. Trong quá trình số hoá người ta biến đổi tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu trong không gian ảnh hai chiều: chiều rộng và chiều cao, các giá trị mẫu là giá trị về độ sáng hay màu của ảnh. Sau khi lấy mẫu ta được ma trận các mẫu của ảnh. Lượng tử hoá ma trận này ta được ảnh số và lưu dưới dạng file. Có rất nhiều loại định dạng ảnh khác nhau, tuy nhiên chúng được chia làm hai loại cơ bản là ảnh bitmap và ảnh vector. Ảnh bitmap lưu trữ theo dạng từng điểm màu được xếp liên nhau theo thứ tự nhất định và có thể nén theo hai phương pháp: nén mất thông tin và nén không mất thông tin. Ảnh vector thực chất là một

tập hợp các đường toán học. Việc lưu trữ ảnh vector là lưu lại các lệnh dùng để vẽ lại ảnh đó. Kích thước của ảnh vector rất nhỏ ($\sim 1/1000$ lần so với ảnh bitmap), chất lượng ảnh không phụ thuộc vào độ phân giải. Có ba cách để chúng ta số hoá ảnh hai chiều đó là sử dụng máy quét, xuất từ đĩa CD hoặc là xuất từ một nguồn bên ngoài, chẳng hạn trên Internet.

Ảnh đồ hoạ 3 chiều là loại ảnh có thông tin màu sắc được mô tả trong không gian 3 chiều. Ảnh 3 chiều chỉ có thể được tạo dựng qua máy tính. Các ảnh 3 chiều không hoàn toàn mô tả chính xác được thế giới tự nhiên, đó chỉ là ảnh vector (là ảnh được tạo bởi từ các hàm toán học), không tồn tại ảnh bitmap 3 chiều. Ngày nay nhờ có phần mềm trợ giúp quá trình tạo ảnh nên chúng ta dễ dàng xây dựng được ảnh chuyển động một cách sinh động. Các phần mềm hiện đang dần hoàn thiện để tạo ảnh gần với ảnh thực hơn, bên cạnh đó lý thuyết về ảnh đồ hoạ 3 chiều vẫn đang dần phát triển không ngừng.

Ảnh đồ hoạ 3 chiều mô tả các đối tượng trong không gian 3 chiều thể hiện qua chiều rộng, chiều dài và chiều cao. Trong đồ hoạ máy tính, vật thể không tồn tại ở hình dạng vật lý. Mọi đối tượng đều biểu diễn qua các công thức toán học và mối quan hệ giữa chúng. Không gian ba chiều trong máy tính là một khối không gian được định nghĩa một cách toán học bên trong máy tính và nó chỉ tồn tại ở bên trong phần mềm. Để tạo được ảnh trong không gian ba chiều người ta phải dựa vào hệ tọa độ 3 chiều được quy định sẵn trong các phần mềm tạo ảnh.

Các hình ảnh 3 chiều được xây dựng từ các điểm ảnh. Một điểm ảnh trong không gian máy tính có 3 tọa độ biểu diễn chiều cao, chiều rộng và chiều sâu của điểm đó. Mỗi tọa độ này được biểu diễn trên một trục riêng tạo thành hệ tọa độ 3 chiều trong không gian máy tính. Trong quá trình tạo ảnh người ta thường cố định một chiều và sửa đổi trên 2 chiều còn lại.

Điểm khác biệt chủ yếu giữa ảnh 2 chiều và ảnh 3 chiều là việc tạo độ sâu cho đối tượng trong ảnh. Các đối tượng 2 chiều được mô tả qua chiều cao và chiều rộng mà không hề có chiều sâu. Mặc dù ở một góc nhìn nào đó trong máy tính, đối tượng 2 chiều giống như đối tượng 3 chiều nhưng khi thay đổi vị trí góc nhìn thì đối tượng 2 chiều sẽ không còn tính chất này nữa. Tuy vậy, các kỹ thuật xây dựng ảnh 3 chiều hiện nay phần lớn lại dựa trên kỹ thuật xây dựng ảnh 2 chiều đã có trước đây.

c. Dữ liệu hoạt hình số - video số

Video là ảnh thực được ghi lại và thể hiện nhờ máy ghi video, phần mềm video và các công cụ phần cứng. Thực chất video là sự kết hợp giữa âm thanh và hình ảnh cùng với sự biến đổi theo thời gian. Bản thân video có tính chuyển động và có sự va chạm giữa các hình ảnh.

Kích thước dữ liệu để tạo tệp video rất lớn nên nhất thiết phải được nén dữ liệu. Tác dụng nén chuyển đổi đoạn video dạng số thành các mô tả toán học chiếm một không gian đĩa ít hơn nhiều so với các hình ảnh theo dạng điểm ảnh. Giải nén sẽ tiến hành công việc theo trình tự ngược lại. Dạng nén thông dụng cho video là MPEG.

d. Dữ liệu âm thanh

Âm thanh có thể là tiếng người, âm thanh nhạc cụ hay âm tổ hợp. Dữ liệu âm thanh được số hoá trên đĩa quang từ, xử lý trên phần mềm âm thanh, xử dụng micro, PC đa phương tiện, bìa âm thanh. Có thể nói bản chất của âm thanh đó là sự dao động không khí. Khi một vật phát ra âm thanh chính là vật đó đã làm không khí xung quanh đó bị dao động. Người ta đo dao động của âm thanh bằng Hz và đơn vị đo độ ồn của âm thanh là dB. Âm thanh giúp con người có thể hiểu nhanh, rõ ràng một vấn đề. Nó khác xa với các tín hiệu từ văn bản bởi vì sự phối hợp giữa âm thanh và hình ảnh giúp cho con người có thể hiểu rõ được mọi sự vật một cách nhanh chóng. Trong tự nhiên khả năng nghe của con người khoảng 40 Hz ~ 44KHz, nếu tần số âm thanh quá cao hoặc quá thấp thì người ta cũng không thể nghe được những âm thanh này. Ta có thể nghe thấy âm thanh có trong thực tế hoặc được con người sáng tạo ra.

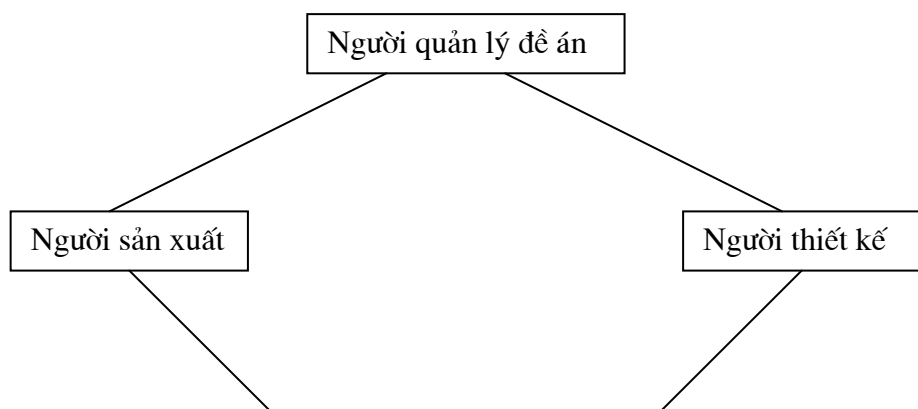
Các tín hiệu âm thanh ở dạng nguyên thể có dạng hình sóng. Trước đây người ta thường thu tín hiệu âm thanh và ghi lại dưới dạng tương tự. Ngày nay, với sự phát triển của công nghệ số nên người ta đã số hoá âm thanh để có thể xử lý tốt hơn cho các ứng dụng thực tế.

Âm thanh trong tự nhiên là sự dao động dạng sóng của khí. Khi được mã hoá dưới dạng sóng điện từ, âm thanh có dạng đồ thị như trên. Để số hoá người ta sẽ lấy mẫu tại các điểm khác nhau dọc theo đồ thị của âm thanh. Số điểm lấy mẫu càng lớn chất lượng âm thanh số càng cao.

5.1.4 Yêu cầu tài nguyên

a. Nguồn nhân lực

Để thiết kế và xây dựng một sản phẩm đa phương tiện thì cần đến: người sản xuất, người quản lý đề án, người thiết kế giao diện, người phát triển, giám đốc nghệ thuật, người lập trình, kĩ sư.



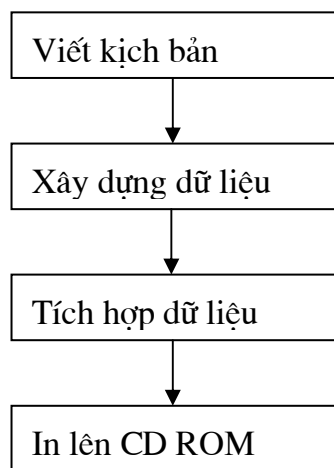
Các vai trò chính trong quá trình sản xuất đa phương tiện.

b. Tài nguyên phát triển sản phẩm đa phương tiện

Như đã đề cập, sản phẩm đa phương tiện là sự kết hợp của âm thanh, hình ảnh, là một chuỗi các sự kiện kế tiếp nhau và người dùng có thể thao tác trên nó. Do đó để tạo sản phẩm đa phương tiện thì chúng ta cần phải có: ổ CD, ổ đọc ghi (VCD), loa và bìa âm thanh, phương tiện MIDI, máy quét, máy ảnh số, máy quay video số, các phần mềm đa phương tiện.

5.2. Pha sản xuất

Bước này có thể coi là khâu chính trong quá trình xây dựng một sản phẩm đa phương tiện, bao gồm các việc:



*Sơ đồ các bước trong quá trình sản xuất
sản phẩm đa phương tiện*

Pha sản xuất đa phương tiện thường bao gồm các ý sau:

- Các yếu tố sản xuất và công cụ: Tùy theo các lựa chọn công cụ phần cứng và phần mềm về đa phương tiện, tùy theo các quyết định về ngân sách có thể, phạm vi của đề án, năng lực của cán bộ.
- Phân vai các bộ phận trong đề án đa phương tiện
- Viết kịch bản: viết thành lời văn, câu chuyện. Câu truyện dưới dạng văn bản dùng ở dạng: văn bản hiển thị, văn bản trên phim bấm. Kịch bản cần phải thể hiện đúng ý người đặt hàng
- Thu thập và xây dựng dữ liệu
- Tích hợp các kiểu dữ liệu
- In lên đĩa CD, viết tài liệu kỹ thuật, tài liệu hướng dẫn sử dụng

Việc số hoá sẽ chuyển các đồ hoạ sang dạng điểm ảnh 2 chiều, đồ hoạ BITMAP, gọi là ảnh RASTER. Các cách số hoá ảnh đồ hoạ:

- Quét
- Nhập vào từ đĩa CD
- Nhập vào từ nguồn khác

Dựa vào nội dung cơ bản trong khâu chuẩn bị sản xuất để ta viết kịch bản. Kịch bản được xem như một bản thiết kế mà từ đó người ta xây dựng nên chương trình. Nói một cách cụ thể thì kịch bản là phương tiện để liên kết dữ liệu (văn bản, ảnh, video, ...) theo nhan đề của sản phẩm. Nó phải chi tiết về câu chuyện và cấu trúc, chi tiết về các sự kiện và những gì xảy ra khi thực hiện. Đồng thời nó phải có tính tương tác và tính thao tác. Có nghĩa là những người phát triển có thể dựa theo kịch bản đó để phát triển sản phẩm khác.

Theo phương pháp viết kịch bản chúng ta sẽ chia làm nhiều đoạn khác nhau, một câu chuyện được chia làm nhiều cảnh. Danh sách các cảnh được tổ chức theo bảng:

số cảnh

tên cảnh

tình huống

Tiếp theo là các sự kiện trong câu chuyện. Các sự kiện được đặt trong danh sách các sự kiện. Danh sách các sự kiện là danh sách tổng hợp các nội dung theo các tình huống, trong đó chưa có giả thiết phục vụ người dùng. Các thông tin về các sự kiện gồm:

- Số cảnh
- Tên sự kiện
- Đích

- Kích hoạt
- Sự kiện
- Kịch tính

Sau khi có tên cảnh, tên sự kiện, chúng ta sẽ tạo giao diện người dùng. Phải sắp xếp vị trí cảnh và các tầng hiển thị. Tiếp theo chúng ta phải lên danh sách các đối tượng có chứa sự kiện. Đồng thời cần phải lên danh sách dữ liệu được sử dụng trong chương trình. Đối với dữ liệu ảnh cần biết kích thước, màu như thế nào. Đối với tệp video thì phải nêu thêm chuẩn nén nó (ví dụ: chuẩn nén là jpeg). Còn đối với dữ liệu âm thanh thì cần đưa ra kiểu và thời gian tiêu tốn... Từ kịch bản người sản xuất sẽ dựa vào đó để xây dựng chương trình.

Dựa vào danh sách dữ liệu được nêu lên trong kịch bản chúng ta sẽ dùng biện pháp thích hợp để thu thập dữ liệu. Có thể sử dụng phần mềm vẽ, các loại thiết bị quay hoặc chụp ảnh số, hoặc cũng có thể sử dụng máy quét, v.v... Dữ liệu cơ bản trong sản phẩm đa phương tiện thông thường là: văn bản, hình ảnh, âm thanh, hình động và đoạn phim.

Tích hợp dữ liệu là bước quan trọng nhất trong pha sản xuất. Sử dụng các dữ liệu đã thu thập được chúng ta sẽ sử dụng phần mềm trộn để tạo sản phẩm đa phương tiện.

5.3. Pha sau sản xuất

Sau sản xuất là bước chỉnh lý sản phẩm theo cơ chế phản hồi về nghệ thuật và nội dung. Có nghĩa là sản phẩm sẽ được đưa ra kiểm thử. Người dùng và người thiết kế sẽ thử nghiệm và đánh giá sản phẩm. Cho người dùng kiểm thử sẽ biết được chương trình đã đạt tiêu chuẩn đến mức độ nào. Người thiết kế kiểm tra để xem xét chương trình có đúng với bản thiết kế hay không. Kết quả cuối cùng sẽ đưa ra sản phẩm hoàn hảo.

Chỉnh lý sản phẩm theo cơ chế phản hồi về nghệ thuật và nội dung. Chọn ra người thử nghiệm, đánh giá sản phẩm. Cho người dùng kiểm tra vì biết rõ yêu cầu, cho người thiết kế kiểm tra để kiểm tra lại mình.

Chuẩn bị các mẫu thử: Mẫu thử cho phép kiểm tra tính khớp của kịch bản với nội dung đặt ra. Cần trình diễn mẫu cho các thành viên nhóm đa phương tiện xem để lấy ý kiến đánh giá bình luận. Nên vẽ cấu trúc điều khiển về đồ họa và âm thanh.

Có nhiều khía cạnh của sản phẩm được phân tích để quyết định sản phẩm đó có ích hay không. Các bước điển hình khi đánh giá sản phẩm là:

- Kiểm tra alpha với nhóm nhỏ
- Bản kiểm tra Beta với nhóm người dùng lớn hơn

- Phản hồi của người dùng đầy đủ

6. Kiến trúc nền cho sản phẩm đa phương tiện

6.1. Nguyên tắc chung xây dựng sản phẩm đa phương tiện

Môi trường học và môi trường thí nghiệm ngày càng phát triển đa dạng. Khi xây dựng một sản phẩm đa phương tiện hỗ trợ giáo dục ta cần tuân theo những nguyên tắc cơ bản sau:

a. Để sử dụng

Thông thường khi người ta không thể thao tác được phần mềm nào đó thì người ta cũng sẽ không sử dụng nó. Sản phẩm đa phương tiện được thiết kế tốt phải cung cấp giao diện nhất quán xuyên suốt tất cả các chức năng của hệ, dễ truy cập và định vị đến nội dung cần thiết, có điểm liên lạc đến nhóm hỗ trợ kỹ thuật và những yêu cầu khác liên quan đến bài học. Ngoài ra, đối với người học sản phẩm cần có công cụ đơn giản trợ giúp họ theo dõi tiến trình và kết quả học tập của mình.

b. Để truy cập

Cần hỗ trợ, mở rộng tính dễ sử dụng đến tất cả người dùng, bất kể trạng thái cơ thể và khả năng vật lý của họ. Tại một số nước phát triển, những khung nền sản phẩm đa phương tiện chính thức thường đòi hỏi nội dung bài giảng phải được truy cập bởi số đông người dùng. Trong một số trường hợp, chính phủ còn đưa ra những quy định bắt buộc như những nguyên tắc và ghi thành luật định chi tiết để các sản phẩm khi xây dựng phải tuân theo.

c. Mềm dẻo

Nói chung một sản phẩm đa phương tiện không nên đưa ra những đòi hỏi, điều kiện đối với người dùng của hệ về cách cơ sở dữ liệu sử dụng hay về cách tương tác của sản phẩm đối với các sản phẩm khác. Người học cần được cung cấp khả năng thao tác trên sản phẩm trong phạm vi rộng nhất có thể của các thiết bị tính toán và truyền thông. Họ có thể điều chỉnh vị trí các đối tượng, thí nghiệm, thay đổi giá trị để có thể quan sát được các thí nghiệm trong các trường hợp khác nhau đặc biệt nhất.

d. Hiệu quả

Một sản phẩm hỗ trợ việc giáo dục đào tạo đòi hỏi có các đặc tính sau: khả năng tương tác qua lại để truy cập nhanh đến nơi lưu trữ dữ liệu và những hệ thống thông tin khác; khả năng tái sử dụng và khả chuyển của các đối tượng thí nghiệm sao cho giảm thiểu đầu tư và tăng khả năng cạnh tranh giữa các sản phẩm; khả năng quản trị được để

tăng hiệu suất của sản phẩm ở mọi lúc; khả năng truy cập của người dùng và khả năng dùng lâu để tránh tình trạng lạc hậu công nghệ.

6.2. Nguyên tắc xây dựng nội dung sản phẩm đa phương tiện

Chất lượng của sản phẩm có vai trò quyết định lớn đến thành công của sản phẩm đa phương tiện dù ở môi trường nào. Nội dung sản phẩm đa phương tiện xây dựng phần mềm mô phỏng hỗ trợ giáo dục cần thoả mãn những yêu cầu sau:

- Tài liệu đào tạo có chất lượng cao
- Tài liệu được soạn bởi các nhà giáo dục có chuyên môn và kỹ năng sư phạm về vấn đề đó
- Được soạn ra bằng cách sử dụng tiến trình thiết kế và công cụ tiên tiến nhất
- Dựa theo những nguyên tắc sư phạm đúng đắn
- Nội dung phải được kiểm tra trước khi tạo thành sản phẩm

a. Tài liệu giảng dạy phải có chất lượng cao

Trước tiên, người soạn bài giảng phải là giáo viên- chuyên gia giỏi trong lĩnh vực đó. Cần phân định rõ mục đích của bài giảng: giới thiệu nội dung hay đi chi tiết thực hành. Từ đó bài giảng sẽ cung cấp những hoạt động cho người học và phát triển những kỹ năng tương ứng.

Việc chuẩn bị tài liệu cho bài mô phỏng thí nghiệm ảo rất khác với việc soạn bài truyền thống và khác với viết sách giáo trình. Bài thí nghiệm điện tử cần được thiết kế chuyên dụng, không đơn thuần như chiếu các tấm slides bài giảng hay phân phát dụng cụ thí nghiệm cho học sinh trong lớp truyền thống. Nội dung bài thí nghiệm cần có tính thời sự, nghĩa là không chỉ giám sát về nội dung mà còn phải để ý đến mọi tình tiết dẫn dắt trong bài thí nghiệm sao cho không lỗi thời, nhất là các ví dụ minh họa, các thí nghiệm liên quan đến thời gian, ngày tháng, công nghệ. Chẳng hạn những dẫn chứng của hai hay ba năm trước nay có thể sẽ không phù hợp nữa. Một lợi thế lớn trong việc chuẩn bị bài thí nghiệm điện tử là khả năng sử dụng lại các tư liệu của thí nghiệm này cho các bài thí nghiệm tương tự hay liên quan khác. Cần lưu ý đặc biệt tính tái sử dụng khi phát triển giáo trình điện tử nói chung và thí nghiệm ảo đa phương tiện nói riêng.

b. Các nhà giáo dục hiểu biết máy tính

Việc chuyển từ cách quan sát thí nghiệm truyền thống sang việc quan sát các hiện tượng xảy ra trong thí nghiệm ảo đòi hỏi những kỹ năng và tiến trình dạy học mới để tạo ra những tư liệu minh họa thích hợp và hỗ trợ người học. Trong các bài giảng thí nghiệm truyền thống thì người dạy có vai trò chủ động nhưng khi có sản phẩm đa phương tiện hỗ trợ thì vai trò đó thuộc về người học. Người soạn nội dung có thể không phải là những

giáo viên truyền thống. Thay vào đó, người soạn phải thiết kế quá trình thí nghiệm được kiểm soát bởi người học. Một người có thể thực hiện những tác vụ phức tạp một cách dễ dàng hơn nhiều so với việc tạo ra một phần mềm trên máy tính làm thay toàn bộ những việc đó. Nói cách khác, người thiết kế thí nghiệm ảo đa phương tiện đang tạo ra các chương trình máy tính gồm những tương tác nhiều phần phức tạp giữa con người và máy tính để hỗ trợ việc học. Để làm được những việc đó thì cần có những kỹ năng mới ở những người phát triển phần mềm.

Việc tạo lập thí nghiệm ảo đa phương tiện đòi hỏi trình độ chuyên môn và kiểm soát chất lượng khác với việc dạy mặt đối mặt truyền thống. Một số nhà giáo dục và đào tạo cảm thấy việc chuyển đổi đó là khó nhọc, vất vả; nhưng cũng có nhiều người lại thấy việc chuyển đổi là hấp dẫn, lý thú và đáng làm.

c. Tiến trình thiết kế và công cụ tiên tiến

Một cách truyền thống, nói chung một bài giảng trên lớp được soạn bởi giáo viên trực tiếp dạy môn đó và phụ thuộc vào phong cách, kỹ năng của giảng viên đó, người khác khó có thể sử dụng với hiệu quả cao như người soạn ra nó và giảng nó. Tuy nhiên ở bài giảng trong phần mềm mô phỏng thí nghiệm ảo lại khác rất nhiều, trước hết là vai trò chủ động thuộc về người học, các tài liệu phải xuất bản điện tử và sẽ được tái sử dụng bởi chính mình và những người khác.

Tiến trình thiết kế cần cung cấp một cách rõ ràng phân định giữa những tác vụ khác nhau trong việc xây dựng bài giảng và phân phát bài giảng, sao cho đội ngũ cán bộ thực hiện có thể phân chia công việc ở các pha/bước khác nhau của quá trình soạn các phần trong bài thí nghiệm. Chẳng hạn cần tách việc soạn nội dung thí nghiệm với việc tạo lập hỗ trợ đa phương tiện, hoặc nên có nhóm chuyên lo về minh họa và thiết kế cho các bài thí nghiệm ảo này.

Tiến trình thiết kế bài thí nghiệm ảo đa phương tiện gồm một số bước như sau:

- **Đặc tả:** xác định mục đích đào tạo của bài thí nghiệm và quyết định những tài liệu nào cần được sử dụng.
- **Thiết kế tổng thể:** lựa chọn thứ tự và kiểu cách biểu diễn, trình bày, đưa vào các bài kiểm tra, đánh giá
- **Thiết kế Module:** quyết định xem nội dung thí nghiệm được thể hiện như thế nào trong từng module
- **Tạo lập module:** dựng các minh họa, đa phương tiện và thiết kế giao tiếp với người dùng
- Tiến trình thiết kế nói trên cũng phải bao gồm tiến trình đảm bảo chất lượng để kiểm tra tài liệu ở từng bước kể trên.

- Việc lựa chọn công cụ trong hệ thiết kế là rất quan trọng. Tùy theo bản chất của thí nghiệm và số lượng bản (phương tiện) sẽ tạo ra, nói chung để tạo ra nội dung học một giờ thí nghiệm (tính chung) UK eUniversities Worldwide cho rằng để soạn bài thí nghiệm điện tử có chất lượng cho một môn học thì hết khoảng 20 người-năm. Vì vậy việc lựa chọn công cụ soạn bài thí nghiệm để tăng năng suất của các tác giả soạn bài là rất quan trọng. Tuy nhiên, công cụ phải hỗ trợ quy trình sư phạm và cơ chế phân phát bài giảng sẽ được dùng sau này.

Một số tiêu chí của công cụ soạn bài thí nghiệm điện tử cụ thể như sau:

- Phải hỗ trợ thích hợp đối với từng bước của tiến trình thiết kế và tạo bài thí nghiệm nói trên
- Phải có mô hình người dùng nhất quán và giao diện người dùng nhất quán cho tất cả các công cụ sao cho một người có thể thực hiện nhiều vai trò trong quá trình thiết kế (chẳng hạn một người có thể là tác giả của môn này, là người đánh giá môn khác, đồng thời là người trợ lý cho môn khác nữa)
- Phải cho phép tác giả sử dụng thời gian hữu hiệu (thao tác nhanh, dễ, chính xác)
- Phải hỗ trợ mềm dẻo việc tạo lập và sử dụng multimedia.
- Phải hỗ trợ những nguyên tắc sư phạm mong muốn
- Phải đi đôi với quy trình đảm bảo chất lượng, chẳng hạn có những công cụ nhúng để xử lý lưu các phần cần ghi nhớ của học sinh.
- Phải tạo ra các tài liệu theo chuẩn mực để cho phép mềm dẻo trong việc sửa chữa và phân phát.
- Phải cho phép tái sử dụng các tài liệu, chia sẻ tài liệu giữa các module và chia sẻ các module giữa các môn học, bài giảng
- Phải hỗ trợ thay đổi tài liệu theo nhu cầu của cơ sở đào tạo.
- Phải hỗ trợ hiệu chỉnh nội dung để đảm bảo tính thời sự của nội dung.
- Phải hỗ trợ thiết kế các bài giảng thí nghiệm điện tử truy cập được để đáp ứng người dùng với những khó khăn về cơ thể và thị lực.

Tuy nhiên, cũng tùy hoàn cảnh cụ thể của từng cơ sở đào tạo, giáo dục từng nước mà những tiêu chí trên có thể được giảm bớt một vài tiêu chí hoặc được bổ sung thêm.

Những nguyên tắc sư phạm đúng đắn đối với sản phẩm đa phương tiện:

Việc giáo dục có sự trợ giúp của công nghệ đã đạt được vị trí tốt hơn dạy và học truyền thống. Muốn vậy, khi soạn bài giảng thí nghiệm ảo người soạn còn phải tuân theo những nguyên tắc sư phạm sau đây:

- **Học tự mình định hướng:** tài liệu và nội dung của bài thí nghiệm phải thiết kế để lôi kéo người học thay vì thúc đẩy giáo viên như tài liệu dạy và học truyền thống, vì vai trò chủ động đã thuộc về người học. Bài thí nghiệm điện tử phải cho phép người học khám phá, duyệt lại đường học của mình, có thể học theo thời gian thích hợp và có thể tìm kiếm những tư liệu liên quan và nhiều hơn nữa. Bài thí nghiệm nhất thiết phải gợi mở những học hỏi, tìm tòi tiếp theo trong tài liệu chính của bài.
- **Tham gia tương tác và giải quyết vấn đề:** bài thí nghiệm phải lôi kéo người học vào quá trình tương tác thay vì giao cho họ phương tiện có chứa bài giảng như đào tạo truyền thống hay đào tạo từ xa lâu nay. Người học sẽ nhớ được thí nghiệm và hiện tượng nếu như họ được yêu cầu phải suy nghĩ sâu về nội dung bài minh họa đó. Một cách giải quyết rất tốt để buộc người học phải suy nghĩ sâu là đặt họ vào trạng thái phải giải quyết vấn đề thay vì chỉ quan sát hiện tượng đơn thuần. Việc lựa chọn vấn đề cũng cần chú ý, vì có thể họ quan tâm sâu sắc, hoặc có thể đẩy họ ra xa nếu quá đơn giản hoặc quá khó (không giải quyết được trong thời gian cho phép). Những vấn đề gợi mở học viên giải quyết phải được tích hợp vào cấu trúc bài thí nghiệm.
- **Thực hành và thử nghiệm kiểm tra:** người học cần được tạo khả năng thử nghiệm với những ý tưởng mới để kiểm nghiệm vấn đề một cách hứng thú. Người học ngồi trước máy tính- phương tiện rất tốt để thực hành, hơn hẳn phương tiện truyền thống là cuốn sách. Việc học sinh kiểm tra, thử nghiệm phần mềm mô phỏng thí nghiệm ảo còn giúp thầy giáo có thể kiểm soát tính tích cực của học sinh, qua đó có thể đánh giá kết quả học tập của học sinh.
- **Sử dụng hợp lý đa phương tiện:** chỉ nên dùng mô hình hóa thí nghiệm bằng video những khi cần thiết, chẳng hạn như những giảng giải cần trực quan thì vấn đề hiển thị, mô phỏng hay âm thanh là khá cần thiết và tăng hiệu quả truyền đạt. Tuy nhiên việc dùng nhiều minh họa cũng tạo ra những ồn ào hay mỏi mắt, phân tán tư tưởng người học.
- **Phản xạ:** sản phẩm đa phương tiện cần cung cấp cho người học khả năng làm trễ thời gian để phản xạ hoặc xử lý với việc quá tải trong nhận thức, trong cả lúc học hay đọc tài liệu bài học và cả lúc tương tác với thí nghiệm, hiện tượng. Phần mềm mô phỏng thí nghiệm ảo cần cho phép tổ hợp cả tương tác thời gian thực lẫn làm trễ thời gian để kịp phản xạ với một sự kiện nào đó trong lúc quan sát hiện tượng.

- ***Phản hồi lại sau:*** người học cần có khả năng có thể truy hồi lại ngay những gì vừa đi qua hoặc sau đó cũng có thể truy hồi lại bất cứ lúc nào, chỗ nào trong bài học. Nghĩa là bài thí nghiệm phải được thiết kế để dùng như tài liệu tra cứu sau này.
- ***Đánh giá và phản hồi người học:*** sản phẩm cần thường xuyên kiểm tra sự tiếp thu và hiểu bài của người học thông qua các bài kiểm tra nhỏ sau mỗi phần thí nghiệm hay bài học. Việc trình bày bài thí nghiệm cần được tu chỉnh cho phù hợp với năng lực và tiến bộ của người học. Vừa phải kiểm tra thường xuyên, vừa phải xử lý kịp thời những phản hồi từ người học về tiến bộ và những vấn đề phát sinh.

Khi phát triển bài thí nghiệm cần lưu ý đến điều kiện và phương tiện của người học và phù hợp cho từng loại đối tượng học. Chẳng hạn khi phát triển cho học sinh phổ thông thì phải chia nhỏ bài học để người học có thể quan sát một vấn đề hay hiện tượng không trải quá dài và lâu. Việc thiết kế bài thí nghiệm có dùng đa phương tiện cũng phải rất chú ý đến văn hóa, tâm lý lứa tuổi và trình độ chung của học sinh để bài thí nghiệm đạt được chất lượng cao.

Năng lực của người học có thể khác nhau về:

- Ngôn ngữ: nếu bài thí nghiệm mô phỏng không được soạn bằng tiếng mẹ đẻ của người học thì cần có bảng từ vựng trực tuyến để người học tham chiếu và sử dụng thuận lợi trong quá trình học
- Trình độ và kiểu hỗ trợ tương ứng
- Mức độ tự vươn lên và mức độ tự giác
- Thời lượng quan sát một hiện tượng và ghi nhớ kết quả thí nghiệm
- Bài giảng cần được soạn sao cho khuyến khích người học tham gia tương tác với các đối tượng trong thí nghiệm.

Trợ giúp phải được cung cấp ở mọi mức, kể cả trợ giúp về tài liệu lẫn trợ giúp về cách sử dụng hệ thống. Sản phẩm đa phương tiện cần cung cấp kịp thời những hỗ trợ trong quá trình học. Các dạng hỗ trợ có thể là:

- Trở lại mục trước, hoặc thể hiện một số tài liệu ở mức sâu hơn
- Truy cập đến các tài nguyên khác
- Hướng dẫn tương tác với các hiện tượng trong thí nghiệm

Bài thí nghiệm cũng cần phù hợp với cơ sở đào tạo. Nội dung bài cần được chia thành các modul và có thể điều chỉnh để cơ sở đào tạo có thể tạo lập những phiên bản cho từng nhóm đối tượng học riêng biệt. Nội dung cũng cần phải được nén chung để các cơ sở đào tạo có thể giải nén và chỉnh sửa dễ dàng.

6.3. Thiết kế giao diện tương tác người dùng máy tính

Giao tiếp người dùng đóng vai trò ngày càng quan trọng trong thiết kế và xây dựng phần mềm đáp ứng yêu cầu ngày càng đa dạng của các lớp người dùng. Nó góp phần làm cho các phần mềm trở nên tiện dụng, hiệu quả và thân thiện hơn. Tuy nhiên việc đánh giá một hệ thống có tính tiện dụng là không đơn giản. Nhiều kỹ thuật đánh giá giao tiếp được nghiên cứu và đề xuất. Tuy nhiên việc áp dụng trong thực tế lại không rõ ràng.

Giao diện sử dụng của một hệ thống thường là tiêu chuẩn so sánh để phán xét hệ thống. Một giao diện khó sử dụng ít ra là sẽ gây nhiều sai lầm của người sử dụng. Một thiết kế giao diện kém có thể làm cho người sử dụng gây ra các sai lầm tai hại. Nếu các thông tin được biểu diễn theo cách lẫn lộn và dễ hiểu nhầm thì người dùng có thể hiểu nhầm ý nghĩa của các khoản mục thông tin và gây ra các thao tác sai. Giao diện người sử dụng phải tính đến nhu cầu kinh nghiệm và khả năng của người sử dụng.

Theo ISO-9241-11, tính tiện dụng của một sản phẩm đa phương tiện thông qua một số người dùng chuyên nghiệp nhằm thực hiện nghiên cứu các mục đích chuyên dụng là ấn tượng, hiệu quả và đáp ứng trong một ngữ cảnh sử dụng riêng biệt.

- Ấn tượng: sự chính xác và đầy đủ, qua đó người dùng thực hiện mục đích chuyên dụng
- Hiệu quả: việc sử dụng tài nguyên một cách tối ưu trong mối quan hệ với tính ấn tượng
- Đáp ứng: tính độc lập với sự thiếu tiện nghi và thái độ tích cực với sản phẩm đó.

Tính tiện dụng không phải là một thuộc tính của sản phẩm riêng lẻ mà là một thuộc tính của việc tương tác với sản phẩm trong ngữ cảnh sử dụng. Một số nhà nghiên cứu đưa ra 5 tiêu chí cụ thể cho tính tiện dụng:

- Tính dễ học
- Tính hiệu quả
- Tính dễ nhớ
- Tính dự đoán lỗi
- Đáp ứng tính chủ quan

Thiết kế lấy người dùng làm trung tâm là một cách tiếp cận phát triển hệ thống tương tác, tập trung đặc biệt vào việc xây dựng các hệ thống tiện dụng. Đó là một hoạt động đa lĩnh vực, tổ hợp các yếu tố con người, các tri thức và các phương pháp kỹ thuật khác. Yêu cầu không thể thiếu đối với một sản phẩm đa phương tiện là có một giao diện tương tác người dùng-máy tính tiện dụng.

7. Đánh giá hiện trạng và lựa chọn kỹ thuật

a. Hiện trạng và xu hướng phát triển

Khi kết nối quá trình học tập với công nghệ máy tính, người ta có thể làm cho mọi học sinh được tiếp cận với nội dung họ cần học một cách trực quan nhất. Ba dạng công nghệ cao cấp riêng biệt đang được tích hợp với nhau là: điện tử dân dụng, viễn thông và vi tính.

Công nghệ điện tử dân dụng với sự sản xuất hàng loạt lớn các thiết bị và linh kiện điện tử với giá thành hạ đã từ lâu được sử dụng trong giáo dục với chức năng là phương tiện nghe nhìn thụ động..Viễn thông cung cấp cho ta khả năng kết nối các vị trí địa lý khác nhau thông qua đường truyền vệ tinh, máy điện thoại. Máy tính điện tử cho ta khả năng có thể thay đổi một cách cơ bản cách thức dạy và học nhờ khả năng nối mạng, lập trình, xử lý logic của nó. Tuy nhiên, chỉ khi có công nghệ vi tính ra đời và phát triển kèm theo các công nghệ nghe-nhìn kỹ thuật số, cùng với sự ra đời của các loại đĩa quang lưu trữ dữ liệu thì khả năng tích hợp cả ba loại công nghệ nói trên đã trở thành hiện thực.

Năm 1991, một thiết bị đầu tiên hội tụ ba công nghệ cao nói trên ra đời tại Los Angeles với tên gọi là Cd-i, viết tắt của kỹ thuật đĩa quang và tương tác. Công nghệ mới này tích hợp ba loại hình công nghệ cao vào trong một công nghệ duy nhất tạo ra một thiết bị. Cd-i được tạo với mục đích xây dựng các đoạn video giải trí có tính tương tác cao. Cd-i là một thiết bị vi tính chuyên dụng tạo khả năng tính toán, suy luận, truy cập nhanh dữ liệu và khả năng tương tác. Thiết bị vi tính này dùng một hệ điều hành đặc biệt tích hợp với một ổ đọc CD có thể truy cập nhanh tới bất kỳ điểm nào trong kho dữ liệu chứa trong đĩa quang tương đương với 10000 trang sách. Kho dữ liệu này gồm đủ dạng: văn bản, âm thanh, ảnh và video. Thiết bị dùng TV để hiển thị mọi thông tin.

Do vậy phần mềm xây dựng sản phẩm đa phương tiện phải có một cơ sở dữ liệu chứa các kiến thức dưới mọi dạng thông tin và một bộ công cụ để tiếp nhận xử lý yêu cầu, mệnh lệnh, truy cập, lập luận và ra quyết định đa thông tin lên bộ phận hiển thị. Tất cả những thành phần trên hợp lại thành một hệ thống phần mềm được lưu trữ trên một đĩa quang CD hoặc DVD rẻ nhất. Như vậy, về kỹ thuật cũng đã được giải quyết. Vấn đề còn lại chỉ là kỹ thuật nội dung, trình diễn sao cho phong phú hấp dẫn, thuận tiện nhưng lại tối ưu về sử dụng bộ nhớ và tốc độ truy cập. Chúng ta thấy những đĩa CD-ROM học tập dùng cho máy tính có thể phổ cập cho người học quen sử dụng PC. Và khi màn ảnh rộng hoặc máy chiếu multimedia trở nên phổ biến thì lúc này việc sử dụng màn hình TV để phổ cập kiến thức cộng đồng là lựa chọn hiệu quả nhất, do đó mọi phần mềm nên thiết kế để có thể hiển thị trên tivi.

Trong khi mọi vấn đề kỹ thuật cho một sản phẩm đa phương tiện đã được giải quyết thì việc tạo ra phần mềm nội dung thích hợp, tận dụng được mọi khả năng của kỹ thuật là điều khó khăn và thiếu thốn nhất hiện nay. Các phần mềm nội dung phải bảo đảm chất lượng nghe, nhìn tối đa, phải có tính tương tác người máy đủ mạnh, truy cập ngẫu nhiên và nhanh đến mọi cơ sở dữ liệu, nội dung phong phú nhưng chiếm ít bộ nhớ và phải có khả năng cập nhật.

Trong quá trình thử nghiệm xây dựng phần mềm và sử dụng các loại phần mềm hiện có trên thị trường Việt Nam chúng ta có các giải pháp lựa chọn kỹ thuật sau:

b. Về kỹ thuật nghe nhìn

Sử dụng Digital Video chất lượng từ trung bình trở lên làm phương tiện nghe nhìn. Hiện nay có khá nhiều định dạng Video. Chúng tôi nhận thấy tối ưu nhất là dùng chuẩn MPEG. Cơ sở của sự lựa chọn này như sau:

- MPEG là chuẩn quốc tế chất lượng cao của hình ảnh động. Là phương thức nén tối ưu dữ liệu âm thanh và hình ảnh động. Mã hóa bằng chuẩn MPEG sẽ làm giảm kích cỡ file video một cách đáng kể. Chuẩn nén và giải nén MPEG2 được sử dụng rất rộng rãi trên thế giới trong truyền hình, truyền vệ tinh, truyền hình cáp kỹ thuật số, DVD và giải pháp cao hơn nữa đã được thực hiện như chuẩn HDTV.
- Nén và giải nén MPEG được sử dụng rộng rãi bởi vì nó có được chất lượng hình ảnh và âm thanh cao hơn với các kỹ thuật tương tự hiện tại. Trong cùng một thời gian thì tốc độ nén của MPEG là rất lớn với hệ số truyền từ 50 tới 1000. Tuy nhiên phương thức nén và giải nén MPEG rất phức tạp cần phải kết hợp nhiều giải pháp kỹ thuật để thu được các thông số chính xác.
- Nên dùng kỹ thuật cứng của SigmaDesigns nếu có thể. Để nén và giải nén dữ liệu MPEG với chất lượng hoàn hảo nó cho phép tạo và phát lại các file nén MPEG với tốc độ truyền dữ liệu là 500 Kb lên tới 15Mb/1s. Tốc độ này được gọi là tốc độ truyền bit, nếu tốc độ truyền bit càng cao thì chất lượng video cũng tăng lên, đồng nghĩa với việc tăng kích cỡ file MPEG.
- Dạng nén MPEG-2 có độ phân giải 720x480, 24 bit (16,8 triệu màu) tốc độ 30 ảnh trên một giây (fps) đối với hệ NTSC và 720x576 tốc độ 25 fps đối với hệ PAL và SECAM. Chuẩn MPEG1 dùng với tốc độ thấp khoảng 1.5MB hoặc thấp hơn có độ phân giải 352x240, 24 bit (16,8 triệu màu) tốc độ 30fps đối với hệ NTSC và 352x288 tốc độ 25fps đối với hệ PAL. Chuẩn MPEG là một chuẩn mở hoàn toàn bắt kịp với sự phát triển của ngành công nghiệp máy tính, truyền thông giải trí với chuẩn hiện thời đang sử dụng cho video số trên cả tivi và máy tính cá nhân.

- Nên sử dụng cả MPEG 1 và 2 để dùng được cho các PC thông thường với Mplayer. Chỉ dùng MPEG2 trong trường hợp rất đặc biệt và khi đó phải sử dụng thiết bị khác để hiển thị.

c. Về kỹ thuật lập trình dùng trong Multimedia

Một trong những yêu cầu của trình diễn nội dung là phải tích hợp được càng nhiều khả năng Multimedia càng tốt. Chúng tôi đã tiến hành các kỹ thuật tích hợp này gồm DigitalVideo và audio, văn bản, hoạt hình, mô phỏng bằng các công cụ khác nhau cũng như viết các chương trình này từ ngôn ngữ bậc cao.

Lập trình với Macromedia Authoware:

Hỗ trợ về mặt giao diện và các kết nối tới video, mô phỏng và xử lý dữ liệu ở các chương trình khác nhau. Đặc biệt khi dùng Authoware thiết kế cho phần giáo dục thì được hỗ trợ rất mạnh về tính chất tương tác ví dụ việc đánh giá, cho điểm các câu hỏi trắc nghiệm và tương tác trở về phần bài giảng tương ứng với câu trả lời sai để học lại một cách dễ dàng giúp người học tiết kiệm thời gian và tập trung sâu vào kiến thức trọng tâm mà mình chưa nắm vững. Để thiết kế cho phần này Authoware đã có xây dựng sẵn các lớp tùy chọn cho phần câu hỏi và câu trả lời cũng như các hàm tính điểm có sẵn chỉ cần có đầy đủ nội dung là thực hiện được ngay. Ngoài ra phần Authoware còn hỗ trợ một số hàm điều khiển dùng cho các thí nghiệm thực được quay bằng video để người dùng có thể tạo các giao diện đẹp hơn trên nền của video và nếu như tốc độ đường truyền cho phép thì có thể ứng dụng viết bằng Authoware đưa lên mạng để học từ xa.

Lập trình mô phỏng bằng Flash:

Đây là chương trình thiết kế cho phần mô phỏng cho hình ảnh đẹp và rất tiện lợi trong quá trình thiết kế, không mất nhiều thời gian vì đã được hỗ trợ nhiều bởi các lớp chuyển động cũng như hiệu ứng sẵn có. Chương trình cũng bổ sung một số hàm đơn giản để thực hiện một số phần tương tác đơn giản trong mô phỏng. Tuy nhiên để thực hiện các mô phỏng có nhiều các tương tác phức tạp thì không thể thực hiện được vì vậy chương trình dùng chủ yếu dùng để mô tả hiện tượng.

Lập trình mô phỏng bằng 3Dmax:

Cũng là một dạng mô phỏng giống như Flash nhưng có thêm hiệu ứng 3D hình ảnh chuyển động giống như hình ảnh thật hơn tuy nhiên khi tạo ra các file chạy thì có dung lượng lớn hơn nhiều so với Flash.

MaxScript thực sự là ngôn ngữ lập trình và ta có thể sử dụng để tự động hóa nhiều vấn đề của MAX, từ việc tạo vật thể cho đến làm hoạt cảnh. MAXScript được thiết kế đặc biệt dành riêng cho MAX và cung cấp các truy cập đến chức năng bên trong của MAX. Với MAXScript ta có thể truy cập hầu như mọi chức năng có sẵn trong giao diện người dùng cũng như một số chức năng không có sẵn khác. Chúng ta có thể tạo ra mọi loại đối tượng có sẵn trong bảng lệnh hoặc có thể truy cập đến đối tượng có thể chỉnh sửa để hiệu chỉnh các thuộc tính của nó. Script là một công cụ rất mạnh nhưng phức tạp trong 3D Studio MAX, chỉ những điều khiển 3 chiều phức tạp người ta mới sử dụng lập trình mô phỏng bằng 3DMAX.

Lập trình Visual Basic:

Đối với các thí nghiệm mô phỏng cần có nhiều các tương tác phức tạp cần phải lập trình bằng ngôn ngữ đầy đủ hơn. VB là ngôn ngữ hỗ trợ nhiều cho lập trình giao diện và các câu lệnh thực hiện cũng đơn giản hơn so với nhiều ngôn ngữ khác. Ngoài ra VB còn được hỗ trợ bởi các hàm API có sẵn nên có thể lập trình trên nền Windows rất dễ dàng. Với thí nghiệm mô phỏng được viết bằng VB thì người dùng có thể thực hiện một thí nghiệm ảo gần giống như một thí nghiệm thật. Các tương tác thực hiện giữa người và các công cụ thí nghiệm đều được thao tác trên máy.

Như vậy, các tương tác khi tiến hành thí nghiệm ảo người dùng tiến hành thực các tương tác đầy đủ như làm thí nghiệm thật, chỉ khác là các đối tượng tiến hành đều là ảo. Tuy nhiên một nhược điểm của thí nghiệm ảo là các hiện tượng vật lý có thể thực hiện không hoàn toàn chính xác như trong quá trình tiến hành với thí nghiệm thực và quá trình thực hiện cần đầu tư nhiều thời gian. Tuy vậy nếu có nhiều thời gian và người lập trình hiểu biết hiện tượng vật lý một cách chính xác thì có thể khắc phục được nhược điểm trên.

Digital Video:

Đối với các thí nghiệm được quay bằng video thì người dùng có cái nhìn trực quan hơn về quá trình tiến hành thí nghiệm nhưng người dùng không thể can thiệp được vào quá trình tiến hành thí nghiệm mà phải theo dõi từ đầu đến cuối nên dễ gây nhàm chán. Ngoài ra thí nghiệm được quay bằng video chiếm dung lượng trên đĩa lớn gấp hàng trăm lần so với thí nghiệm ảo.

Macromedia Director:

Macro Director là công cụ đầu tiên trên thế giới để tạo ra các sản phẩm Multimedia. Director thực sự hữu ích bởi nó cung cấp cho người thiết kế tạo ra những sản phẩm phức

tập, những ứng dụng tương tác bằng cách sử dụng sức mạnh của ngôn ngữ lập trình Lingo. Macromedia Director đã thoả mãn nhu cầu về nghệ thuật lập trình cho nhiều người, nó là một sản phẩm duy nhất đủ dễ và khá trực quan cho nhiều người có cơ hội tạo ra một ứng dụng ấn tượng ngay lần đầu tiên họ sử dụng chương trình. Sức mạnh của nó đủ để các nhà phát triển kết hợp được những ứng dụng multimedia tinh vi với giao diện 3D, truy cập cơ sở dữ liệu, và kết nối internet lại với nhau.

Tóm lại, không có một kỹ thuật lập trình hoặc công cụ hỗ trợ nào đủ để thoả mãn mọi yêu cầu tạo ra phần mềm mô phỏng thí nghiệm ảo đủ mọi yêu cầu. Việc kết hợp các ưu điểm của các loại lập trình dùng trong lập trình Multimedia sẽ giúp ta có bộ cơ sở dữ liệu sinh động dùng trong giáo dục hay các vấn đề khác có liên quan. Dựa vào các tính năng trên ta có thể chọn Director làm chương trình chính cho việc thiết kế giao diện, kết nối, tạo tương tác giữa các hiện tượng mô phỏng khác nhau trong thí nghiệm ảo.

Phần thí nghiệm thật được quay bằng video để người dùng có cái nhìn trực quan hơn, phần này cũng được kết nối bằng Director kết hợp thêm phần điều khiển bên ngoài để có thêm nhiều tính năng và giao diện trông đẹp hơn. Tuy nhiên nó sẽ bị hạn chế bởi dung lượng rất lớn.

Phần mô phỏng là phần rất quan trọng trong kỹ thuật Multimedia. Đối với các hiện tượng đơn giản chỉ cần dùng để mô tả hiện tượng có thể dùng Flash hoặc 3DMax còn đối với các mô phỏng cần có nhiều tương tác phức tạp ta dùng VB để thiết kế. Phần này thì có thể thực hiện được rất nhiều thí nghiệm mô phỏng bởi dung lượng cho các mô phỏng này rất nhỏ.

8. Lập kế hoạch

Cần có kế hoạch rõ ràng cho từng người thực hiện trong mỗi bước sản xuất và có thời hạn kết thúc công việc. Việc tổ chức lại người thực hiện công việc là điều nên tránh (do phải đào tạo lại, do quan hệ những người cùng làm một công việc). Công việc chỉ được giao cho người khác thực hiện khi cần tổ chức lại:

- Tăng, giảm nguồn nhân lực do thay đổi kịch bản
- Kỹ năng cán bộ yếu
- Tiến trình chậm, rủi ro, đứt đoạn vì công việc khác,
- Thay đổi ekip, quan hệ các thành viên có vấn đề.

Do vậy cần phân công công việc, trách nhiệm cho từng thành viên tham gia xây dựng sản phẩm một cách rõ ràng, cân đối. Những người thực hiện ngoài công việc của mình còn phải có trách nhiệm với những người khác cùng làm công việc liên quan. Đặc thù của sản phẩm đa phương tiện có tính hình thức cao do vậy những người thực hiện cần

tuân theo sự thống nhất trên kịch bản trước đó và cố gắng hoàn thiện theo đúng tiến độ thực hiện vì các giai đoạn xây dựng sản phẩm có quan hệ mật thiết với nhau. Nhà quản lý thay đổi kế hoạch thực hiện dựa vào tiến độ công việc của những người thực hiện thông qua báo cáo về tiến độ công việc của họ.

9. Nội dung thực hiện

9.1. Cấu trúc sản phẩm

Trong quá trình xây dựng, đạo diễn và nhà sản xuất cần tạo khung cấu trúc phần mềm cơ bản nhất (bao gồm những phần chính gì, các đặc trưng và chú ý đặc biệt trong các phần đó). Chỉ khi có cấu trúc phần mềm chúng ta mới có thể xây dựng kịch bản chi tiết và xây dựng dữ liệu cho chương trình.

9.2. Viết kịch bản

Dựa vào nội dung cơ bản trong phần lập kế hoạch chúng tôi viết kịch bản chi tiết. Kịch bản được xem như một bản thiết kế chi tiết, dựa vào đó để xây dựng chương trình. Kịch bản liên kết dữ liệu (văn bản, ảnh, video, ...) theo chủ đề chính của sản phẩm. Người phát triển có thể dựa theo kịch bản này để phát triển sản phẩm multimedia khác.

Sau khi có tên cảnh, tên sự kiện, chúng ta sẽ tạo giao diện người dùng. Chúng ta lên danh sách các đối tượng có chứa sự kiện và danh sách dữ liệu được sử dụng trong chương trình. Các thành viên sẽ dựa vào kịch bản để xây dựng chương trình.

Bảng chi tiết kịch bản cho từng đối tượng đa phương tiện:

- Cấu trúc phác họa từng trang (hình ảnh)
 - Mô tả tổ chức logic các cảnh và mối quan hệ giữa chúng
 - Mô tả các đối tượng dữ liệu tương ứng xuất hiện trong cảnh
- Phác họa cấu trúc chi tiết (văn bản)
 - Nội dung dữ liệu chi tiết trong mỗi cảnh tương ứng
- Phác họa chi tiết từng trang
 - Thiết kế các phần bổ sung (không lấy đi khuôn dạng) chi tiết hơn (màu sắc, kích cỡ, kiểu chữ, màu nền, sắp đặt hình ảnh và kích thước, hoạt hình, video, audio)
 - Mức độ tương tác với người học

Một kịch bản phải thể hiện được các đề xuất sản phẩm thực hiện. Tạo cảnh như thế nào, nó có mô tả được nội dung ý chính cần trình bày không, nó có thể hiện được đúng theo sơ đồ logic đã trình bày không, các đối tượng điều khiển có được đặt cố định tại một vị trí nào không. Khi kịch bản hoàn thành thì người sản xuất có thể hoàn toàn dựa vào đó để xây dựng nên sản phẩm đa phương tiện.

Đối với sản phẩm mô phỏng thí nghiệm ảo đa phương tiện thì màn hình thể hiện thường có kích thước 800x600 pixel. Hình thức sử dụng chương trình là các mối liên kết giữa màn hình này với màn hình kia. Trong mỗi trang có thể có chứa hình ảnh video minh họa thí nghiệm.

1.1. Cài đặt phần mềm

Thông thường một sản phẩm mô phỏng thí nghiệm ảo đa phương tiện cần sử dụng các phần mềm sau:

- Các phần mềm văn phòng, soạn thảo văn bản như MS OFFICE
- Phần mềm hỗ trợ đa phương tiện (DIRECTOR, AUTHORWARE)
- Phần mềm chỉnh sửa VIDEO (ADOBE PREMIERE, VIEAD VIDEO STUDIO)
- Các phần mềm soạn thảo đồ họa (PAINTSHOP PRO, PAINT, DESIGNER, PICTURE PUBLISHED)
- Các phần mềm soạn thảo 3 chiều (BRYCE 3D, INFINI-D, D4.5, 3D MAX, RAYDREAM, DESIGNER, MAYA)

1.2. Yêu cầu phần cứng

1- Thiết bị

Máy MP MOTOROLA hoặc máy MP INTEL

Cần có SERVER, PRINTER, SCANNER, SOUND CARD.

2- Máy tính đa phương tiện

Máy mức 1: 386SX/2MRAM/30M HDD có loa

Máy mức 2: 486SX/25 MHZ/ 8MRAM/CD và loa

Máy mức 3: Pentium 75MHZ/ 8MRAM

Nói chung, một sản phẩm đa phương tiện được tạo ra cần có các thiết bị sau:

- Máy tính

- Loa và bìa âm thanh
- Máy ảnh số
- Máy quay video số
- Thiết bị chuyển đổi tín hiệu DV300 và card 1394 chuyển đổi tín hiệu từ máy quay video qua máy tính.
- CD Rewriteable

1.3. Xây dựng kho dữ liệu đồ họa cơ bản

Tiến hành xây dựng kho dữ liệu các yếu tố đồ họa số hóa cơ bản (tĩnh và video) phục vụ cho việc phát triển các thí nghiệm mô phỏng

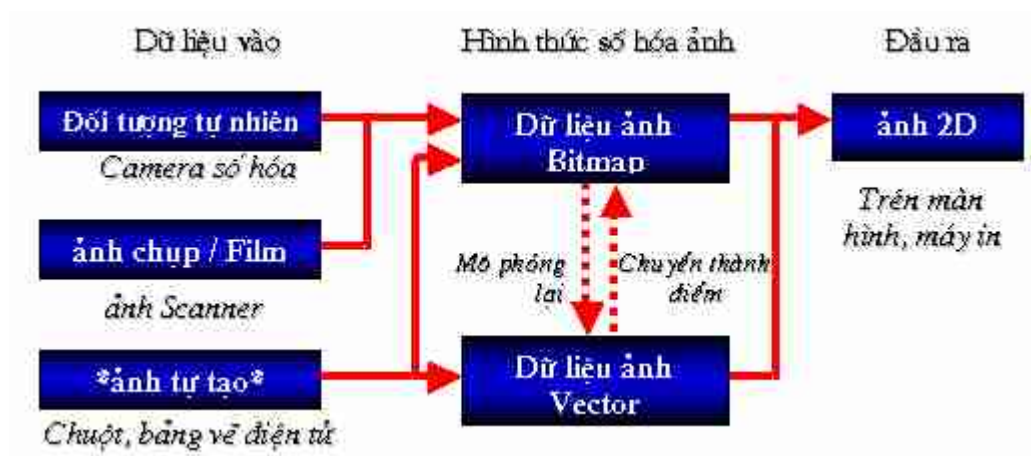
1.3.1 Xây dựng dữ liệu văn bản

Khi xây dựng dữ liệu văn bản cần chú ý:

- Tài liệu dạng văn bản được sử dụng theo đúng định dạng trong sách giáo khoa có bài thí nghiệm cần mô phỏng.
- Nội dung được phân chia theo hiện tượng, thí nghiệm
- Các đề mục đồng mức được sử dụng cùng định dạng về phông chữ.

1.3.2 Xây dựng dữ liệu ảnh

Xây dựng kho dữ liệu ảnh 2 chiều thông qua hình thức số hóa ảnh 2 chiều và định dạng chuẩn các hình ảnh đó để có thể dùng được trong các ứng dụng khác nhau. Hình thức số hoá ảnh 2 chiều được thể hiện qua sơ đồ sau:



Một số định dạng ảnh Bitmap thông dụng:

- BMP
- GIF
- JPEG
- PNG
- PICT(ảnh tạo trong máy Macintosh)
- PCX
- TIFF
- PSD (Adobe Photoshop)

Một số phần mềm soạn thảo ảnh Bitmap thông dụng:

- Adobe Photoshop
- Corel Photo-Paint
- Jasc Paint Shop Pro
- Micrografx Picture Publisher
- Ulead PhotoImpact
- Microsoft Paint

Một số định dạng ảnh Vector thông dụng:

- AI (Adobe Illustrator)
- CDR (CorelDRAW)
- CMX (Corel Exchange)
- CGM Computer Graphics Metafile
- DRW (Micrografx Draw)
- DXF AutoCAD
- WMF Windows Metafile

Một số phần mềm soạn thảo ảnh Vector thông dụng:

- Adobe Illustrator
- CorelDRAW
- Macromedia Freehand
- Xara
- Serif DrawPlus

Trong quá trình xây dựng ảnh phục vụ mô phỏng thí nghiệm ảo đa phương tiện chúng ta cần sử dụng rất nhiều các định dạng ảnh khác nhau, bảng liệt kê dưới đây cho chúng ta cái nhìn tổng quan nhất để xây dựng được các hình ảnh thích hợp.

a. GIF

GIF (từ viết tắt của Graphics Interchange Format) là một kiểu định dạng ảnh bitmap ra đời năm 1987 bởi hãng CompuServe, sử dụng thuật toán nén không mất dữ liệu (thông tin) LZW. Ảnh GIF có tối đa 256 màu (8 bits), được đánh chỉ số index trong bảng màu, cho phép tạo màu nền trong suốt.

b. BMP

BMP là một định dạng ảnh chuẩn bitmap trong môi trường Microsoft Windows. Ảnh được lưu có chỉ số hay không chỉ số các dữ liệu điểm ảnh RGB (tính theo pixel) với độ lớn (kích thước 1, 4, 8, hay 24 bit) một cách có hiệu quả. Dữ liệu lưu có thể không nén hay nén theo thuật toán nén dữ liệu 4-8 bit RLE. BMP là sự lựa chọn tốt nhất đối với các định dạng ảnh bitmap đơn giản có hỗ trợ phạm vi dữ liệu điểm ảnh RGB rộng.

c. JPEG

JPEG viết tắt của Joint Photographic Experts Group, một tổ chức tạo ra định dạng ảnh đồ họa. JPEG sử dụng thuật toán nén mất thông tin. Ảnh JPEG có tối đa 16 triệu màu (24-bit), cung cấp thông tin định rõ tỷ lệ, mức độ nén, cho phép lưu lại từng phần trên trang Web gọi là một "progressive JPEG."

d. TIF và TIFF

Loại file ảnh bitmap này sử dụng thuật toán nén không mất dữ liệu và chúng ta có thể ghi lại dưới các định dạng file khác nhau, TIF và TIFF được sử dụng rộng rãi. Photoshop cho phép chuyển đổi TIF sang định dạng ảnh vector và có thể lưu tới 4 kênh màu. Với ảnh đồ họa trên web, chúng ta không thể thực hiện được 3 kênh màu sử dụng trong chế độ màu RGB (độ sâu màu là 24 bit).

e. PNG

PNG (Portable Network Graphics) là một định dạng ảnh đồ họa bitmap được phát triển thành định dạng ảnh chuẩn bởi W3C. PNG cho phép chúng ta sử dụng tới 256 chỉ số màu, hỗ trợ 16~256 triệu màu (24- 48 bit màu), hỗ trợ việc tạo và lưu alpha kênh, tạo ảnh nền trong suốt, đặc trưng riêng của ảnh PNG là thuật toán nén không mất dữ liệu (thông tin), tốt hơn 10 ~ 30 % so với ảnh GIF.

f. SVG

Là định dạng ảnh vector được tích hợp từ hai định dạng PGML và VML(Precision Graphics Markup Language và Vector Markup Language) là một định dạng ảnh vector được công nhận trong hội nghị World Wide Web Consortium (W3C), một chuẩn chính cho các file ảnh trên Web bởi các tổ chức Adobe Systems, IBM, Netscape, và Sun Microsystems.

g. SWF

SWF (Shockwave Flash) là 1 định dạng ảnh đồ hoạ vector của hãng Macromedia Flash. SVG và SWF đều là định dạng ảnh đồ hoạ vector được dùng phổ biến hiện nay, tất cả đều được mô tả bởi các phương trình, đường toán học mà không phải là tập hợp các điểm ảnh pixel. Đặc trưng của loại ảnh này là kích thước nhỏ hơn rất nhiều so với các định dạng ảnh khác, ảnh vector này cho phép điều chỉnh, phóng to mà không ảnh hưởng tới chất lượng ảnh (độ phân giải, màu sắc, tạo răng cưa hoặc bất kỳ một hiệu ứng nào khác). Cả SVG và SWF đều được đưa ra thảo luận và được công nhận bởi W3C. Hai chuẩn này đều dựa trên chuẩn XML (Extensible Markup Language).

h. EPS

Là ảnh vector được sử dụng thông dụng nhất hiện nay thay thế cho ảnh bitmap hay các đối tượng vector khác. Khi gửi ảnh tới một máy không có định dạng font chuẩn thì chúng ta nên chuyển đổi dòng text sang dạng ảnh vector này.

i. So sánh một số định dạng file ảnh đồ hoạ

GIF

Ưu điểm:

- Được phổ biến rộng rãi nhất
- Hỗ trợ tạo mức hiệu ứng nền trong suốt transparency
- Có khả năng tạo chỉ số trong bảng màu
- Nén không mất dữ liệu (thông tin)
- Thích hợp cho các ảnh có số màu hiển thị không quá nhiều như ảnh đen trắng, ảnh có từng vùng màu đơn như truyện tranh, đường kẻ nghệ thuật.

Nhược điểm:

- Số màu tối đa sử dụng là 256 màu
- Không thể thay đổi, điều khiển được tỷ lệ nén
- Vì số màu được dùng hạn chế như thế nên không phù hợp với ảnh chất lượng cao, ảnh nhiều màu đa dạng trong tự nhiên

JPEG

Ưu điểm:

- Có nhiều tỷ lệ nén mất dữ liệu khác nhau
- Có thể nén với tỷ lệ nén rất cao
- Cung cấp 24-bit màu (16 triệu màu)
- Định dạng tốt nhất đối với các ảnh cần nhiều màu sắc tự nhiên (màu liên tiếp nhau trong bảng màu) như ảnh chụp phong cảnh, ảnh và các minh họa trên sách báo, ảnh 3 chiều.
- Có thể tải từng phần ảnh trên mạng vào từng thời điểm khác nhau (quá trình "progressive JPEGs")

Nhược điểm:

- Không thể đánh chỉ số màu trong bảng màu
- Không hỗ trợ tạo ảnh nền trong suốt
- Nén mất dữ liệu
- Không phải mọi máy tính đều hiển thị tốt, cố định 24-bit màu

PNG

Ưu điểm:

- Cho phép tạo 256 chỉ số màu
- Hỗ trợ 24 ~ 48 bit màu
- Hỗ trợ việc tạo, lưu alpha channel và tạo nền trong suốt
- Đặc biệt tỷ lệ nén không mất dữ liệu tốt hơn GIF từ 10 ~ 30 %

Nhược điểm:

- Không phải là chuẩn được sử dụng rộng rãi, các trình duyệt Web chưa hoàn toàn hỗ trợ định dạng file này
- Gần đây chỉ mới được sử dụng như là một định dạng file có thể đọc được bởi trình duyệt Netscape và IE, điều đó có nghĩa rằng chỉ một phần nhỏ những người sử dụng 2 trình duyệt Web này mới đọc được ảnh PNG

VECTOR (PGML, SWF và VML)

Ưu điểm:

- Kích thước nhỏ hơn nhiều so với các file định dạng ảnh bitmap
- Có thể điều chỉnh tỷ lệ, kích thước, phóng to, thu nhỏ mà không ảnh hưởng tới chất lượng ảnh
- Rất thích hợp cho ảnh có các hình, đường toán học cơ bản, các biểu tượng, bản đồ, đồ họa
- Ngày càng hứa hẹn sẽ trở thành một chuẩn được phổ biến rộng rãi và phát triển mạnh trên Web

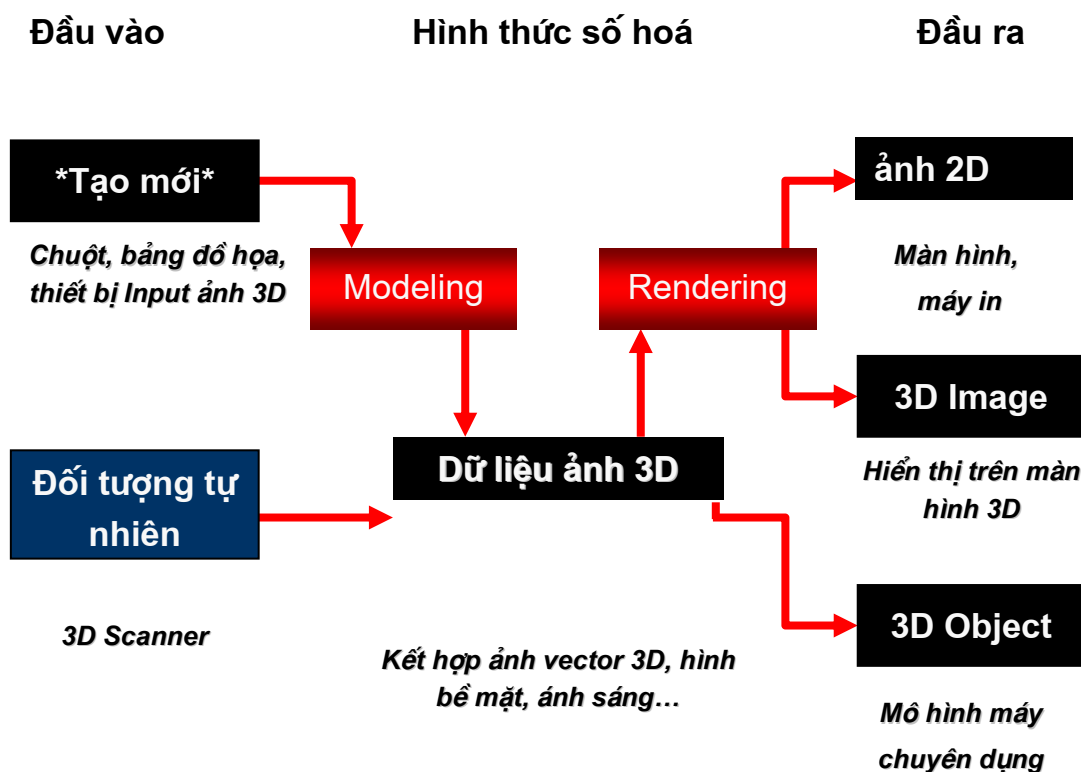
Nhược điểm:

- Hiện nay, chưa tạo thành chuẩn được người dùng sử dụng nhiều trên trang Web
- Đòi hỏi phải dùng hỗ trợ đặc biệt trong hầu hết các trường hợp hiển thị (trừ định dạng SWF)

Và khi lưu vào hệ quản lý cơ sở dữ liệu đa phương tiện chúng ta cần theo các chú ý sau:

- Số ảnh: trong từng bước thực hành đều có ảnh minh họa
- Tên ảnh: tất cả tên ảnh được đánh số như sau: ima + số thứ tự bài + số thứ tự của ảnh trong bài
- Kích thước file ảnh: tối đa 100K
- Định dạng ảnh: *.jpeg, .gif, .png
- Màu sắc: 2-16 triệu màu

Hình ảnh 3 chiều được xây dựng dựa trên hình thức số hóa ảnh 3 chiều:



Mô hình hoá ảnh 3 chiều là một khối hình học được xây dựng đại diện cho một đối tượng nào đó trong không gian 3 chiều. Khối hình học được xây dựng và quản lý trong máy tính là một vật thể được tạo thành từ nhiều vật thể thứ cấp cơ bản. Xây dựng đối tượng 3 chiều từ chuột và đối tượng 2 chiều là rất phức tạp. Mỗi phương pháp tạo mô hình quản lý các đối tượng có những ưu điểm và nhược điểm riêng. Một đối tượng có thể được xây dựng bằng nhiều phương pháp mô hình hóa khác nhau. Người ta có nhiều phương pháp để xây dựng bất kỳ một đối tượng nào dựa trên:

1. Các nhóm đối tượng cơ bản dạng nguyên thủy
2. Các đối tượng quay và đối xứng qua một trục
3. Hình lưới
4. Xây dựng đối tượng từ các tham số bề mặt
5. Các đối tượng đặc biệt đặc trưng trong các lĩnh vực khác nhau
6. Từ phương pháp tính khác ...

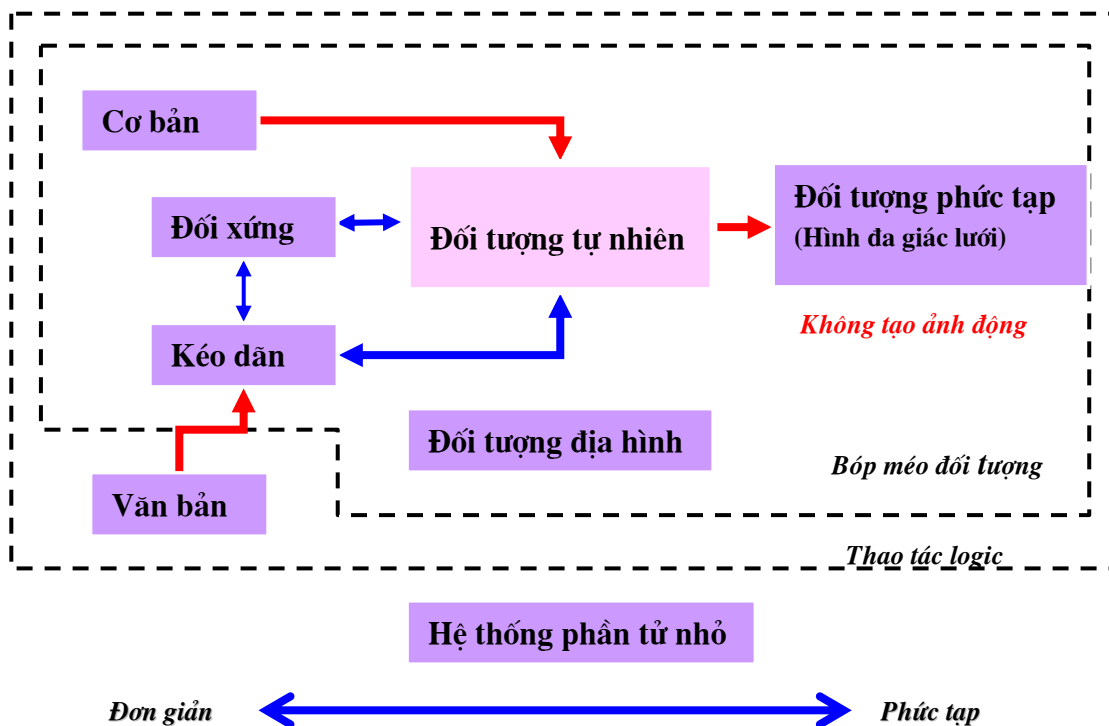
Mọi phần mềm tạo ảnh đồ họa 3 chiều đều phải thực hiện tuần tự theo 7 bước cơ bản sau:

Tạo đối tượng mới



7. Chỉnh sửa đối tượng
8. Tạo bề mặt, kết cấu bề mặt đối tượng
9. Chọn ánh sáng, điều chỉnh nguồn sáng chiếu
10. Đặt vị trí góc quay camera
11. Tạo ảnh chuyển động
12. Đặt thuộc tính Rendering ảnh 3 chiều

Phương pháp mô hình hóa tạo ảnh 3 chiều:



Trong quá trình xây dựng ảnh 3 chiều, người ta thường tạo vật thể từ các đối tượng cơ bản đã được tích hợp sẵn trong chương trình bởi các ưu điểm sau:

- Quá trình Rendering nhanh
- Ảnh xây dựng mịn, bóng, có thể tạo các đối tượng trong ảnh một cách cân xứng, hài hoà
- Tạo đối tượng phức tạp khác từ đối tượng cơ bản bằng cách sử dụng các phép toán logic

Một số đối tượng cơ bản tạo nên ảnh 3 chiều:

- Đối tượng đối xứng: Phương pháp tạo đối tượng đối xứng được xây dựng dựa vào thuật toán Bezier (thuật toán sửa đổi đường cong). Chúng ta cần chú ý tới việc áp dụng các phép toán logic lên các đối tượng chồng gối lên nhau. Cho tới nay, một số phần mềm xử lý ảnh 3 chiều vẫn chưa có khả năng sửa đổi đối tượng có hình dạng phức tạp một cách hoàn chỉnh. Khi đó người ta thường phải nhập đối tượng phức tạp dưới dạng file EPS từ các phần mềm tạo đối tượng ảnh vector khác (*Adobe Illustrator/Corel Draw*)
- Đối tượng dạng văn bản: Các đối tượng dạng văn bản có thể kéo dẫn, thu nhỏ, xoay, xoắn từng ký tự. Quá trình sửa đổi dòng văn bản 3 chiều cũng tương tự như ta sửa đổi trên phần mềm tạo văn bản 2 chiều khác. Tuy vậy sự bóp méo dòng văn bản lại không dễ dàng được thực hiện, chúng ta phải chỉnh sửa từng ký tự trước khi tạo hiệu ứng cho cả dòng
- Đối tượng đối xứng qua trục: Đối tượng có trục đối xứng: vùng cắt ngang là hình tròn, có thể thay đổi hình dạng dọc theo trục thẳng. Vùng cắt ngang có hình dạng bất kỳ, khi kéo dẫn hình dạng đối tượng không thay đổi. Trục là đường thẳng đứng hoặc là đường bất kỳ.
- Đối tượng phức tạp dạng mạng lưới: Là đối tượng được biểu diễn dưới dạng hình học chi tiết nhất nhưng rất phức tạp khi sửa đổi vì khi cần sửa một điểm trên đối tượng thì các điểm cạnh đó thay đổi theo. Đối tượng mạng lưới được xây dựng từ một tập đa giác riêng lẻ, do đó ta có thể sửa đổi chỉ trên một hình đa giác nào đó. Trước tiên người ta thường sửa theo trục đối tượng cơ bản, sau đó mới chuyển đổi thành đối tượng mạng lưới. Ngoài ra, người ta còn xây dựng mô hình đối tượng *Meta-ball* trong một số phần mềm khác và nhập vào dưới dạng file DXF. Nhược điểm cơ bản của đối tượng dạng mạng lưới là chúng ta không thể tạo được ảnh chuyển động như khi xây dựng với đối tượng khác. Đối tượng dạng mạng lưới luôn là đối tượng tĩnh.
- Đối tượng địa hình: Mô tả chi tiết thông tin chiều cao đối tượng trên mặt phẳng 3 chiều.

Song song với mỗi phần mềm tạo ảnh 3 chiều được ứng dụng và phát triển là một định dạng ảnh ra đời với nhiều thuộc tính đa dạng và phức tạp. Mặc dù có nhiều định dạng ảnh đồ họa 3 chiều nhưng cho tới nay vẫn chưa có một định dạng chuẩn chung nhất, có nhiều định dạng ảnh mang tính chất độc quyền của nhà sản xuất. Tùy theo phần mềm mà chúng ta phải định dạng ảnh 3 chiều cho phù hợp.

➤ Chuẩn công nghiệp:

3- DXF (Drawing Exchange Format)

4- 3DF Autodesk

➤ Chuẩn cho hệ máy Macintosh:

5- 3DMF (3D Meta file)

➤ Chuẩn quốc tế:

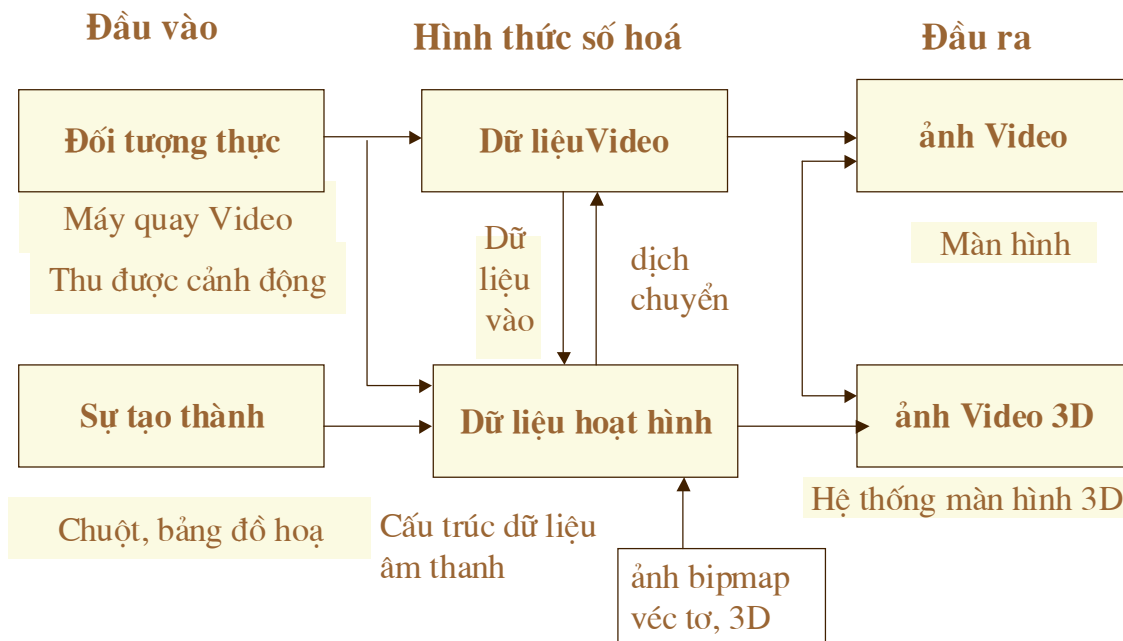
6- VRML (Virtual Reality Markup Language)

7- VRML 97

8- Web 3D - ISO

1.3.3 Xây dựng dữ liệu hoạt hình

Phương thức số hóa dữ liệu hoạt hình:



Tạo và số hoá ảnh động, công việc này bao gồm cả mô hình hoá 3 chiều và hoạt hình. Hoạt hình 2 chiều biểu diễn các đối tượng theo trục X và Y; hình 3 chiều dùng thêm trục thứ ba Z. Hoạt hình thường là sản phẩm sáng tạo của một nhóm, không phải là sản phẩm của cá nhân.

Bản chất cảnh hoạt hình là cảnh động, có sự ảnh hưởng lẫn nhau, có các tính chất như cảnh Video thật, cảnh hoạt hình thường có nhiều sự va chạm,

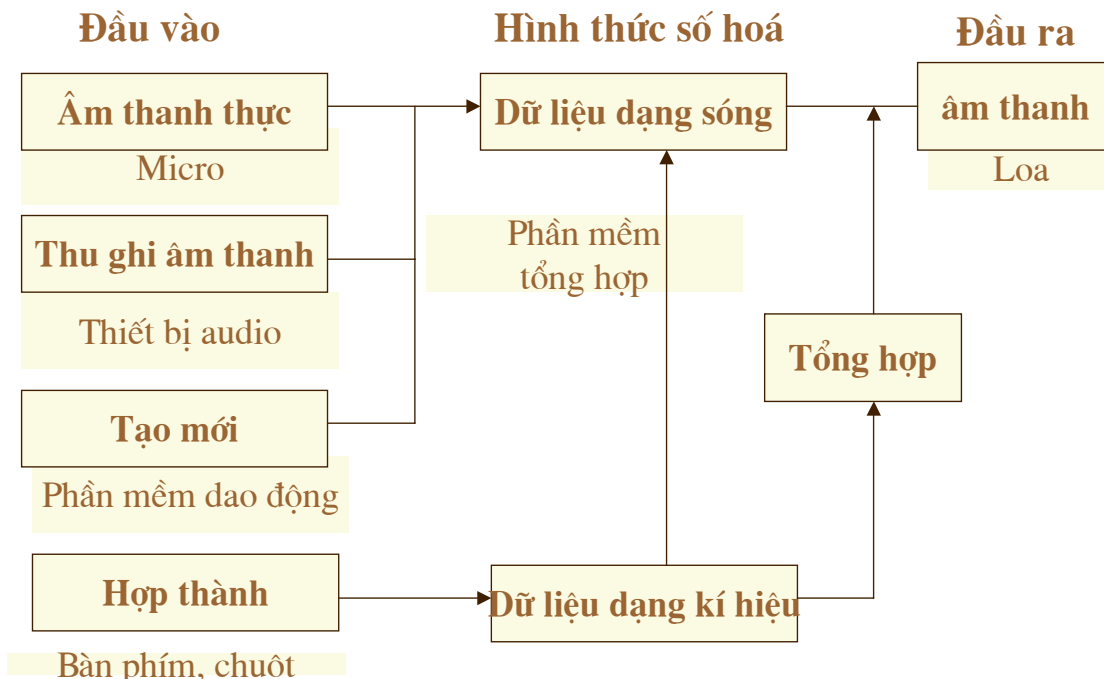
Các tham số hoạt hình:

- Tạo hướng chuyển động trong phim hoạt hình

- Tạo Morphing: thay đổi dần dần từ ảnh này sang ảnh khác, là sự hoà hợp giữa trạng thái biến đổi và pha trộn màu sắc
- Véc tơ hoạt hình: dễ dàng mô tả cảnh động. Để thay đổi vị trí, độ phóng to, thu nhỏ, cảnh quay vòng chỉ việc đặc tả bằng tham số. Ngoài ra, đối với cảnh chính thì không cần thiết vẽ từng khung cảnh.
- Cảnh hoạt hình 3D: áp dụng hầu hết các công nghệ tiên bộ trong công nghiệp phim ảnh. Chuyển đổi cảnh động thông qua mô hình giới hạn bộ khung vật lý. Các cảnh hoạt hình 3D hợp thành uyển chuyển sẽ tạo thành Video.

1.3.4 Xây dựng dữ liệu âm thanh

Hình thức số hoá âm thanh được thực hiện:



Theo sơ đồ trên chúng ta thấy đầu vào của âm thanh có từ rất nhiều nguồn khác nhau (có thể là âm thanh thực có trong tự nhiên). Thông qua các thiết bị thu như micro chuyển hóa âm thanh thành dạng sóng điện từ và ghi vào băng đĩa. Chúng ta cũng có thể dùng các thiết bị sao chép âm thanh như đầu video, radio cassette,... để chuyển âm thanh từ băng, sang băng, từ băng sang đĩa,... Chúng ta cũng có thể tạo âm thanh bằng cách xây dựng các bộ dao động như các thiết bị âm nhạc... Với sự hỗ trợ của các phần mềm tổng hợp âm thanh chúng ta có thể tạo âm thanh từ các ký hiệu. Ví dụ: chúng ta có thể chơi nhạc bằng

bàn phím, xây dựng một bản nhạc bằng cách soạn các nốt nhạc sau đó cho phát lại trên máy tính.

Từ các nguồn âm thanh khác nhau các âm thanh này đều được chuyển hoá thành sóng điện từ và được số hoá. Các dữ liệu sau khi được số hoá sẽ được máy tính xử lý. Sau đó các dữ liệu này sẽ được chuyển ngược thành âm thanh thực thông qua hệ thống loa.

Âm thanh cho video là ứng dụng phổ dụng nhất của âm thanh trong đa phương tiện. Nhiều phần mềm cho phép ghi âm thanh trực tiếp vào máy tính, qua micro, băng và CD như phần mềm ADOBE PREMIERE. Các yêu cầu về phần cứng âm thanh là:

- Loa giám sát
- Micro
- Bìa âm thanh/ bìa đa phương tiện

Định dạng âm thanh

Các định dạng âm thanh được sử dụng phổ biến khi xây dựng sản phẩm đa phương tiện:

- Tập tin âm thanh WAV: đây là dữ liệu âm thanh dạng sóng (waveform audio), dựa trên nguyên tắc số hóa sóng âm. Âm thanh từ một nguồn phát sóng âm (micro, cassette,...) sẽ được chuyển thành dạng tín hiệu số (digital form). MPC (Multimedia PC) lưu chúng trên bộ nhớ hay tập tin.WAV trên đĩa. Các tín hiệu số này có thể thông qua phần cứng biến đổi lại thành âm thanh.
- Tập tin âm thanh MIDI: MIDI (musical instrument digital interface - giao diện số với nhạc cụ) là một chuẩn quen thuộc trong lĩnh vực âm nhạc điện tử. Vào những năm 80, một số nhà sản xuất nhạc cụ điện tử đã phát triển một nghi thức gọi tắt là MIDI để các nhạc cụ điện tử khác nhau (synthesizer - bộ tổng hợp, controller - bộ điều khiển) có thể trao đổi được với nhau. Và từ đó máy tính đã giao tiếp được với nhạc cụ điện tử. Khác với âm thanh dạng sóng, nơi mà âm thanh thực sự được số hoá và phát lại, MIDI chỉ lưu lại những thông điệp (message) điều khiển bộ tổng hợp phát ra âm thanh. Vì vậy mà kích thước của tập tin MIDI nhỏ hơn rất nhiều so với tập tin.WAV (chỉ bằng 1/10 kích thước của cùng file WAV).
- Tập tin âm thanh MP3: MP3 (Movie Picture Experts Group- Layer 3) là chuẩn âm thanh dạng nén với công nghệ cao, được tạo ra từ các chuẩn MPG (chuẩn nén hình video). MP3 cũng là file âm thanh dạng sóng như WAV nhưng nhờ công nghệ lọc bỏ tạp âm mà chất lượng âm thanh của nó khá cao. Kích thước của tập tin MP3 cũng chỉ bằng 1/10 kích thước của cùng file WAV.

Hiện nay có nhiều phương pháp nén âm thanh khác nhau và được chia thành 2 loại:

- Nén không mất thông tin: thuật ngữ nén không mất thông tin ở đây được hiểu theo nghĩa là mọi file âm thanh nén đều được giải nén thành chính âm thanh gốc đã được nén trước đó:
 - Nén dạng entropy với tỉ lệ: 1.5 ~3.0
 - Nén kiểu *LPAC* với tỉ lệ: 1.5 ~4.0

Đây là chuẩn nén không mất thông tin dạng sóng 8 bit, 16 bit, 20 bit hoặc 24 bit (âm thanh đơn hoặc đa kênh) được hỗ trợ trong hầu hết các hệ điều hành: Windows, Linux và Solaris. Nó sử dụng thuật toán CRC đảm bảo quá trình xử lý, truyền phát không mất thông tin. Quá trình mã hóa nhanh trong thời gian thực (4x-12x trên máy 500 MHZ Pentium).

Nén không mất thông tin là ưu điểm chính của định dạng file LPAC so với các định dạng file âm thanh nén mất thông tin thông dụng hiện nay như MP3, WMA, RealAudio. Ngược lại, việc sử dụng thuật toán nén mất thông tin cho ta tỷ lệ nén âm thanh rất cao. MP3 với tốc độ 128 kbit/s có tỷ lệ nén là 11 trong khi LPAC chỉ đạt tỷ lệ nén từ 1,5 đến 4 và phụ thuộc hoàn toàn vào dữ liệu âm thanh. Ví dụ như LPAC có tỷ lệ nén là 2 cho âm thanh dạng nhạc pop và 2,5 cho loại âm nhạc cổ điển. Khi sử dụng định dạng nén này rất có thể chúng ta sẽ nhận được hoàn toàn âm thanh dạng bit đơn trong quá trình nén và giải nén file âm thanh. Hầu hết các định dạng nén nguyên thủy không mất thông tin khác như Zip, LZH, Gzip đều có tỷ lệ nén là 1 (hoàn toàn không nén được file âm thanh)

LPAC được sử dụng trong trường hợp file âm thanh cần đạt chất lượng tốt nhất trong quá trình phát mà định dạng MP3 không đáp ứng được. Các định dạng file LPAC có đuôi là .PAC được xây dựng không mất thông tin và tương thích với mọi hệ điều hành cũng như bất kỳ quá trình xử lý âm thanh nào.

Nén kiểu AAC (Advanced Audio Coding) có tỉ lệ nén: ~14lần được sử dụng trong MPEG-2/4

Người ta coi AAC là định dạng nén âm thanh có chất lượng tốt nhất trên Internet hay trên các đường truyền băng thông rộng. AAC được sử dụng rộng rãi trong các máy hát tự động và các thiết bị âm nhạc khác. Không những thế, AAC còn được coi là cơ sở hạ tầng trong việc truyền phát dữ liệu âm thanh trên Internet. Hãng Liquid Audio dự định phát triển một kỹ thuật tiên tiến nhất trong việc xử lý âm thanh để tích hợp vào AAC trong năm tới.

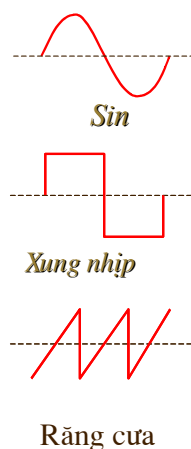
So sánh với MP3 người ta thấy rằng kỹ thuật AAC đã giảm tới 30% không gian lưu trữ dữ liệu. Kỹ thuật AAC đạt được điều này do đã loại trừ được tới 90% tín hiệu âm thanh gốc mà không hề ảnh hưởng tới chất lượng của âm thanh đó. AAC đã chính thức trở thành định dạng chuẩn quốc tế về âm thanh như các chuẩn kỹ thuật MPEG-2 hay MPEG-4.

Đối với dữ liệu âm thanh dạng sóng chưa được xử lý: người ta sử dụng định dạng WAV. Đây là định dạng dữ liệu dạng sóng được sử dụng trong môi trường Windows. AIFF là định dạng Audio dùng trong các hệ máy Macintosh, Amiga, Silicon Graphics.

Đối với dữ liệu âm thanh dạng sóng đã được nén: đặc trưng nhất của dữ liệu âm thanh này là định dạng MP3. MP3 có chất lượng cao, đồng thời tỷ lệ nén tốt. Nó được sử dụng nhiều trong các ứng dụng phát hành trên Internet.

AAC là kỹ thuật mã âm thanh dùng cho việc phát hành và phân phối các sản phẩm âm nhạc. Kỹ thuật AAC cho chất lượng nén cao. Các kiểm chứng độc lập nhau về hiệu quả của quá trình nén và giải nén cho thấy AAC hơn hẳn các định dạng âm thanh khác như MP3 hay bất kỳ mã nén âm thanh trực giác nào khác. AAC cung cấp 48 kênh âm thanh, và tốc độ lên tới 96kHz.

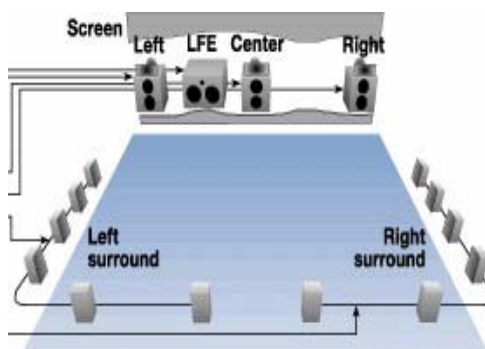
Tạo âm thanh:



Người ta có thể tạo được các dạng âm thanh nhân tạo bằng cách xây dựng âm thanh dựa trên các đồ thị của các hàm toán học. Tạo âm thanh dạng sóng hình Sin, hình xung nhịp, hình răng cưa...Việc tạo âm thanh nhân tạo này được ứng dụng trong rất nhiều trong các chương trình trò chơi giải trí. Âm thanh dạng kí hiệu cơ bản có nghĩa là ứng với một khoảng mức tần số âm thanh nào đó thì người ta mã hoá và chuyển thành một ký hiệu như các nốt nhạc (đô, rê, mi, fa, son, la, si ...) Nó có đặc điểm dữ liệu âm thanh không chính xác, chỉ mang tính giải thích logic chúng ta có thể thay đổi âm thanh bằng cách thay đổi cường độ, thời gian, vận tốc.. Chất lượng âm thanh phụ thuộc vào thiết bị đầu ra. Đặc

trung của loại này là kích thước dữ liệu nhỏ $\sim 1/1000$ so với dữ liệu dạng sóng. Âm thanh ở dạng này thường có định dạng MIDI. Nó được ứng dụng trong các nhạc cụ điện tử.

Âm thanh 3D thực:



Trước tiên muốn có được âm thanh 3D cần phải có một thiết bị ghi, thu đặc biệt. Thiết bị này sẽ thu âm thanh theo nhiều kênh khác nhau và ở các góc độ khác nhau. Cách mô phỏng hay phát lại âm thanh người ta thường sử dụng hàm chuyển HRTF. Hiện nay người ta ứng dụng rất rộng rãi hàm HRTF để mô phỏng và tạo âm thanh 3D từ âm thanh 2D. Tạo âm thanh 3D được ứng dụng nhiều trong các trò chơi máy tính, hệ thống nhà hát nhỏ, hộp từ xa.

Trong sản phẩm đa phương tiện âm thanh được xây dựng dựa trên nguyên tắc:

- Mọi dữ liệu âm thanh dạng sóng đều sử dụng nguyên tắc số hoá sóng âm dạng tập tin WAV.
- Các tín hiệu số này có thể được biến đổi, lọc và nén bằng một số phần mềm thông dụng để tạo thành âm thanh chuẩn hơn
- Tần số âm thanh: 11500 HZ

Đối với hệ thống Window người ta thường sử dụng các kiểu nén âm thanh sau:

- Intel Audio Software codec: kiểu nén này sử dụng cho nhạc và lời nói trên Internet. Khả năng nén tối đa là 8:1. Mã nén này được thiết kế để làm việc với chương trình Intel Video Software.
- TrueSpeech: kiểu nén này sử dụng cho nói chuyện trên mạng Internet với tốc độ truyền thấp.
- Microsoft GSM 6.10: sử dụng cho telephone trên Internet ở châu Âu.

- MS-ADPCM (Microsoft implementation of Adaptive Differential Pulse Code Modulation) kiểu nén này sử dụng để tạo các tệp audio có chất lượng cao ghi trên đĩa CD-ROM. Kiểu nén này được sử dụng rộng rãi.
- Microsoft IMA ADPCM: kiểu nén này được sử dụng tạo các tệp audio cho các sản phẩm multimedia. Kiểu nén này dựa trên mã nén ADPCM được phát triển bởi IMA(Interactive Multimedia Association).
- Lucent Technologies SX8300P: kiểu nén này sử dụng cho giao tiếp trên Internet tốc độ thấp
- elemedia TM AX2400P: kiểu nén này được sử dụng tạo các tệp âm nhạc chất lượng cao trên Internet
- Voxware Audio Codecs: kiểu nén này sử dụng cho các tệp âm thanh dạng tiếng nói trên Internet tốc độ thấp

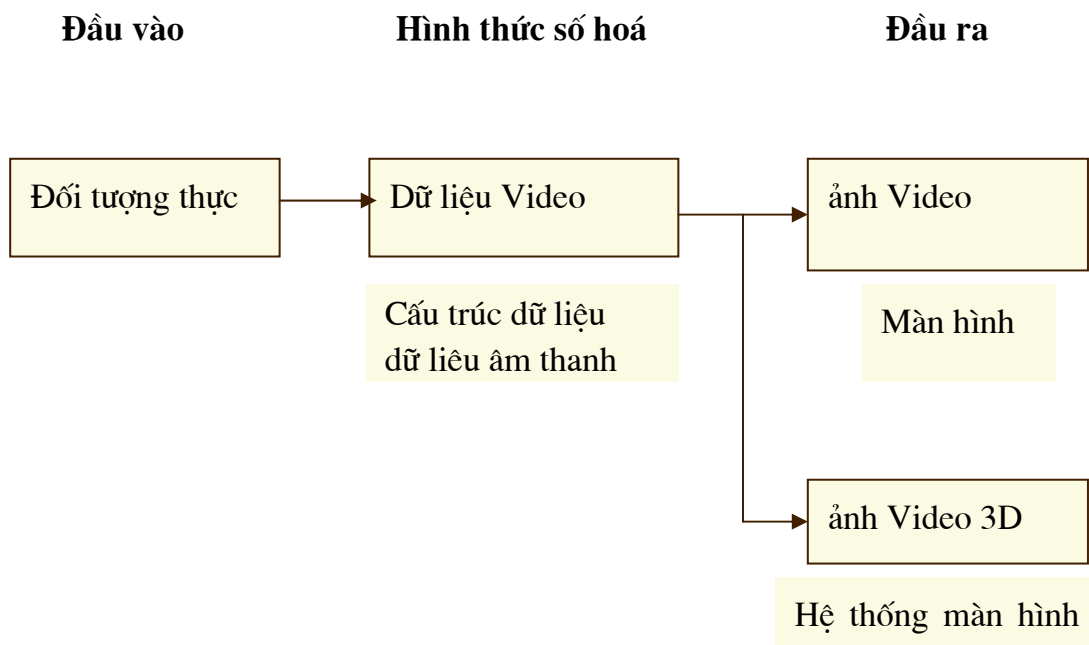
Đối với hệ thống Macintosh người ta thường sử dụng các kiểu nén âm thanh sau:

- mLaw 2:1 : kiểu nén này được sử dụng cho việc chuyển các tệp audio để sử dụng tốt với các ứng dụng trên các máy trạm. Tại đó mLaw là định dạng audio chuẩn . mLaw được sử dụng cho telephone số ở Bắc Mỹ và Nhật Bản.
- 16-bit Big Endian và 16-bit Little Endian: kiểu nén này được sử dụng cho các phần cứng và phần mềm chuyên dụng nhưng thông thường nó không tốt cho việc soạn thảo.
- 24-bit Integer và 32-bit Integer: kiểu nén này được sử dụng cho audio 24-bit hoặc 32-bit nguyên. mã nén này được sử dụng tốt cho các phần cứng và phần mềm chuyên dụng nhưng thông thường nó không tốt cho việc soạn thảo.
- IMA 4:1: kiểu nén này được sử dụng cho các tệp audio tại các trạm. IMA 4:1 được phát triển bởi IMA sử dụng mã ADPCM.
- 32-bit Floating và 64-bit Floating: Các kiểu nén này được sử dụng trong các thiết bị phần cứng và phần mềm chuyên dụng nhưng thông thường không sử dụng cho soạn thảo Video.
- ALaw 2:1: giống mLaw nhưng được sử dụng cho telephone số ở Châu Âu.
- QDesign Music Codec: Sử dụng để nén các tệp audio chất lượng cao sử dụng trên Internet. Chất lượng của nó tương đương các tệp âm thanh trên CD-ROM có định dạng 16-bit, 44.1 kHz. Các tệp âm thanh được nén theo kiểu này có thể nghe trực tiếp bằng đường Internet có tốc độ 28.8 Kbps.
- Qualcomm PureVoice: Đây là kiểu nén được thiết kế tạo các tệp âm thanh tốt nhất ở tần số 8 kHz. Cơ sở của kiểu nén này dựa trên thuật toán nén chuẩn của Telephone CDMA (Code Division Multiple Access).

- MACE 3:1 and MACE 6:1 :kiểu nén này để tạo các tệp audio thông thường cho hệ điều hành Macintosh. Nó dựa trên thuật toán nén âm thanh MACE (Macintosh Audio Compression and Expansion codec). Kiểu nén MACE 3:1 có tỷ lệ nén thấp hơn kiểu nén MACE 6:1 nhưng có chất lượng cao hơn

1.1.1 Xây dựng dữ liệu video

Hình thức số hóa dữ liệu video:



Vấn đề quan trọng nhất ảnh hưởng đến chất lượng ảnh video và thời gian nén là quá trình đánh giá và bù chuyển động. Mặc dù hiện nay đã có các sản phẩm nén video số theo tiêu chuẩn MPEG, nhưng các yêu cầu về chất lượng truyền thông video thời gian thực vẫn chưa thoả mãn được yêu cầu đa dạng của các ứng dụng thực tiễn. Hiện nay giải thuật đánh giá chuyển động nhanh đã được các nhà khoa học trên thế giới nghiên cứu và phát triển rộng rãi trong những năm gần đây nhằm cải thiện chất lượng ảnh video và thời gian nén. Các giải thuật được đề cập theo các hướng: giảm số điểm đối sánh, giảm số pixel tính sai khác, khai thác đặc tính tương quan khối kẻ.

Nén tín hiệu video số là một vấn đề đang được quan tâm nghiên cứu rộng rãi. Các phương pháp mã hóa dựa trên kỹ thuật mã dự báo và mã biến đổi là các kỹ thuật phổ biến nhất được chấp nhận bởi các chuẩn mã hóa video như H.261-263 của IUT-T.

Phương pháp mã hóa được sử dụng chủ yếu trong mô hình nén video của MPEG là mã dự đoán và mã biến đổi. Các phương pháp mã biến đổi nhằm khai thác đặc tính dư thừa không gian trong mỗi ảnh tĩnh. Đây là các kỹ thuật nén chung áp dụng cho mọi loại dữ liệu. Các phương pháp mã dự đoán khai thác đặc tính dư thừa thời gian giữa các ảnh liên tiếp. Kỹ thuật được sử dụng chủ yếu trong mã dự đoán là kỹ thuật đánh giá và bù chuyển động. Trong đó, các kỹ thuật đối sánh khối đã được sử dụng phổ biến do tính đơn giản và hiệu quả. Đối sánh khối thực hiện phân chia mỗi frame ảnh thành các khối có kích thước phù hợp và tiến hành tìm kiếm ở frame tham chiếu khối phù hợp nhất nhằm tạo ra vector chuyển động xác định vị trí dịch chuyển của khối trong frame tham chiếu.

Ảnh Video (ảnh động) là một chuỗi các frame ảnh xuất hiện liên tiếp tạo cảm thụ chuyển động theo thời gian. Tốc độ xuất hiện của các frame ảnh và độ phân giải của các ảnh là các nhân tố quan trọng ảnh hưởng đến chất lượng video. Các chuẩn video hiện nay hầu hết sử dụng khuôn dạng theo chuẩn của kỹ thuật truyền hình (NTSC, PAL) với cấu trúc dữ liệu phân cấp nhằm tăng hiệu quả mã hóa và đảm bảo chất lượng ảnh.

Phương pháp nén video số bao gồm các vấn đề về cấu trúc số hóa giảm tốc độ dòng bit, nén ảnh tĩnh giảm độ dư thừa thông tin theo không gian (mã hóa intra frame) và kỹ thuật đánh giá chuyển động giảm độ dư thừa thông tin theo thời gian (mã hoá inter frame)

Chuẩn MPEG (Moving Picture Expert Group) là chuẩn mã hóa và nén tín hiệu video-audio. Chuẩn MPEG theo ISO và IEC, được sử dụng trong lưu trữ và truyền thông đa phương tiện. Các chuẩn thông dụng hiện nay: MPEG-1, MPEG-2, MPEG-4, MPEG-7

Chuẩn nén video của MPEG được xây dựng chủ yếu trong MPEG-1 và các giải pháp cao hơn được phát triển trong MPEG-2 và các chuẩn sau này. Các nguyên lý cơ bản được áp dụng xuyên suốt trong quá trình phát triển các chuẩn. Mục tiêu chính của phần MPEG video là xác định khuôn dạng có thể sử dụng để mô tả dòng bit video đã được mã hóa. Dòng bit tín hiệu video này là tín hiệu đầu vào của quá trình mã hóa mà nó sẽ nén thông tin ảnh video một cách đáng kể. Tuy nhiên nó xác định việc mã hóa dòng bit này thế nào để tạo điều kiện cải thiện chất lượng ảnh và giảm thời gian mã hóa.

Video dạng MPEG (nhóm ảnh động chất lượng cao):

	MPEG1	MPEG2	MPEG4
<i>Miêu tả</i>	Chất lượng TV/ video	Chất lượng Video nhà hát	Phân chia lớp cơ bản Video
<i>Thuật toán nén</i>	DCT, nén động	DCT, nén động	Nén đối tượng cơ bản
<i>Loại màn hình</i>	352 x 240	720 x 480	Bất kỳ
<i>Loại tỉ lệ</i>	1.2 Mbps	5~20Mbps	~ 64Mbps
<i>Đặc trưng</i>	-	+	+ đối tượng media tác động lẫn nhau
<i>Audio</i>		+AAC, nhiều kênh	+ ghép đôi VQ
<i>ứng dụng</i>	Video CD	DVD	

Trong Window người ta thường sử dụng các kiểu nén Video sau:

- Microsoft RLE : Kiểu này dùng để nén các frame có kích cỡ lớn và màu phẳng (các ảnh không có chiều sâu), ví dụ: để làm các phim hoạt hình. Kiểu nén này có mã độ dài 8 bit dùng thuật toán nén không mất thông tin RLE(Run –Length-Encoding). Chất lượng video cao.
- Microsoft Video1: Dùng cho nén video dạng tương tự (analog video). Kiểu mã nén này hỗ trợ các điểm có 8bit, 16 bit độ sâu.
- Indeo (R) video R3:2 : Sử dụng để nén video 24 bit dùng cho đĩa CD. Kiểu nén này có tỷ lệ nén tốt hơn, chất lượng tốt hơn, và tốc độ hiển thị (khi xem video) nhanh hơn so với kiểu nén Microsoft Video1. Cho kết quả tốt nhất nếu sử dụng mã nén Indeo Video trên dữ liệu video mà trước đó dữ liệu không bị nén với tỷ lệ cao. Khi sử dụng loại dữ liệu này để hiển thị lại thì chúng ta có thể so sánh các mã nén này với kiểu nén Cinepak.
- Cinepak code by Radius: Sử dụng để nén video 24 bit dùng cho CD-Rom hoặc Web. Đây là kiểu nén đạt được tỷ lệ nén cao hơn và tốc độ phát lại (giải nén) nhanh hơn so với kiểu nén video 1. Chúng ta có thể đặt chất lượng hình ảnh để có thể hiển thị lại video tốt với tốc độ 30KBps. Mã Cinepak được hiển thị lại rất nhanh nhưng khi nén mất rất nhiều thời gian. Nó không phù hợp cho việc soạn thảo video mà chỉ phù hợp cho việc chuyển một đoạn video thành kết quả cuối cùng.

- Intel Indeo 5.10 được sử dụng cho các định dạng Video phân tán trên mạng Internet cho các máy tính có bộ xử lý MMX or Pentium II. Đây là kiểu nén có đặc trưng: lựa chọn nén nhanh, mềm dẻo. Kiểu nén này cho phép người xử lý video có thể điều chỉnh việc hiển thị video đối với các băng thông khác nhau. Ví dụ có thể điều chỉnh để video có thể download với modem 56KB, 28,8KB hay đường cáp mạng... Mã nén này được thiết kế để làm việc phù hợp với mã Intel Audio Software.
- Intel Indeo Video Raw R1.1: được sử dụng tốt nhất cho việc thu các thông tin Video ở dạng dữ liệu không nén. Nó làm việc cùng với thiết bị Intel video-capture cards. Mã nén này cung cấp các hình ảnh chất lượng cao. Các tệp video dùng lựa chọn này có kích cỡ nhỏ hơn các tệp không dùng lựa chọn vì ở kiểu này màu sắc đã được chuyển từ model RGB thành model YUV .
- Intel Indeo Video Interactive: đây là kiểu nén tương tự như định dạng 5.10 nhưng nó có một số hỗ trợ cho các đặc tính trong suốt, nhiều đối tượng chuyển động.. trong video. Nó được hỗ trợ bởi các phần mềm tiện ích của Intel.

Trong Macintosh người ta thường sử dụng các kiểu nén Video sau:

- Component video: được sử dụng cho thu video, lưu trữ video hay tạo các đoạn video trung gian (tạm thời). Kiểu nén này có tỷ lệ nén rất thấp do đó chiếm rất nhiều không gian đĩa
- Graphics: được sử dụng cho việc nén các ảnh chất lượng cao với độ sâu màu 8 bit. Mã nén (Graphics codec) này thường sử dụng cho các ảnh tĩnh nhưng đôi khi cũng sử dụng cho việc nén video bởi vì mã nén này không đạt được tỷ lệ nén cao. Nó thích hợp cho các tệp video được chạy (lưu trữ) trên đĩa cứng chứ không phù hợp với các tệp video đặt trên CD-ROM
- Video: được sử dụng cho việc thu và nén các tín hiệu video có nguồn ở dạng Analog. Mã nén này cho kết quả cao khi xem lại các tệp video được lưu trữ trên ổ đĩa cứng. Cho chất lượng vừa phải nếu xem trên CD-ROM. Nó hỗ trợ cả hai loại nén: nén theo không gian và nén theo thời gian cho video 16 bit. Dữ liệu có thể nén lại hoặc dịch lại sau khi nén với tỷ lệ cao hơn mà chất lượng không suy giảm.
- Animation: được sử dụng cho việc nén các khuôn hình có vùng màu có kích cỡ lớn. Ví dụ: các khuôn hình cho phim hoạt hình. Mã nén màu cho phép thay đổi tỷ lệ nén. Với tỷ lệ nén là 100%, video không bị nén. Nếu tỷ lệ dưới 100% tệp video bị nén ở dạng mất thông tin. Mã nén Animation dựa trên lý thuyết nén của Apple và thuật toán nén RLE.
- Motion JPEG and Motion JPEG: được dùng cho mục đích chuyển mã video ví dụ như chuyển các tệp video trong máy tính, các đoạn video trên băng ra các thiết bị khác

của máy tính như đĩa CD ... thông qua thiết bị thu video (video-capture card). Các mã nén này được hỗ trợ nhiều trong các chip có trên các thiết bị thu video như video-capture card do đó tốc độ xử lý rất nhanh.

- Photo-JPEG: kiểu nén này được dùng để nén các ảnh tĩnh có màu sắc biến đổi dần (các đường biên không rõ nét). Đây là kiểu nén mất thông tin nhưng có thể đạt được tham số nén để ảnh có chất lượng rất cao. Mã nén Photo-JPEG là kiểu nén đối xứng theo thời gian nhưng thời gian nén rất lâu. Các ảnh đã được nén theo kiểu này thì không nên dùng làm nguồn để soạn thảo vì nó đã bị mất thông tin. Tuy nhiên nó có tỷ lệ nén cao và chất lượng ảnh tốt nên có thể dùng để lưu trữ hoặc để di chuyển giữa các hệ thống máy tính.
- H.263: Sử dụng tạo các video cho hội thảo. Mã nén này có tỷ lệ nén thấp. Không nên sử dụng chuẩn này cho soạn thảo video thông thường.
- DV-PAL and DV-NTSC: sử dụng mã này tạo video số theo chuẩn PAL và NTSC. Mã nén này dùng để tạo các tệp video với định dạng PAL, NTSC phục vụ in ra băng theo các hệ trên hoặc ngược lại lấy từ băng vào máy tính thông qua digital-video capture card. Chuẩn nén này rất hữu dụng cho việc chuyển dữ liệu video từ hệ thống máy tính này sang hệ thống khác hoặc từ thiết bị này sang thiết bị khác.
- Cinepak: được sử dụng để nén video 24 bit. Các tệp sử dụng kiểu nén này để dùng cho CD-ROM và Web video. Mã nén này có tỷ lệ nén cao và tốc độ giải nén nhanh. Cinepak dùng thuật toán nén không đối xứng các tệp video có kích cỡ nhỏ nhưng thời gian nén rất lâu. Cho kết quả tốt nhất nếu dùng mã nén này để tạo tệp video kết quả.
- Sorenson Video: được sử dụng để nén video 24 bit. Các tệp sử dụng kiểu nén này để dùng cho CD-ROM và Web video. Nó cũng giống như kiểu nén Cinepak nhưng đây là kiểu nén mới thiết kế để nén với chất lượng cao. Mã nén này cho hình ảnh tốt hơn, kích cỡ tệp video nhỏ hơn so với kiểu Cinepak vì vậy nó phù hợp cho việc tạo các tệp video cuối cùng chứ không phù hợp cho soạn thảo.
- Planar RGB: mã nén này được sử dụng hiệu quả cho các khuôn hình có vùng màu đặc như các tệp Animation. Nó sử dụng thuật toán nén RLE kết hợp với kỹ thuật tạo mã animation (Animation codec).

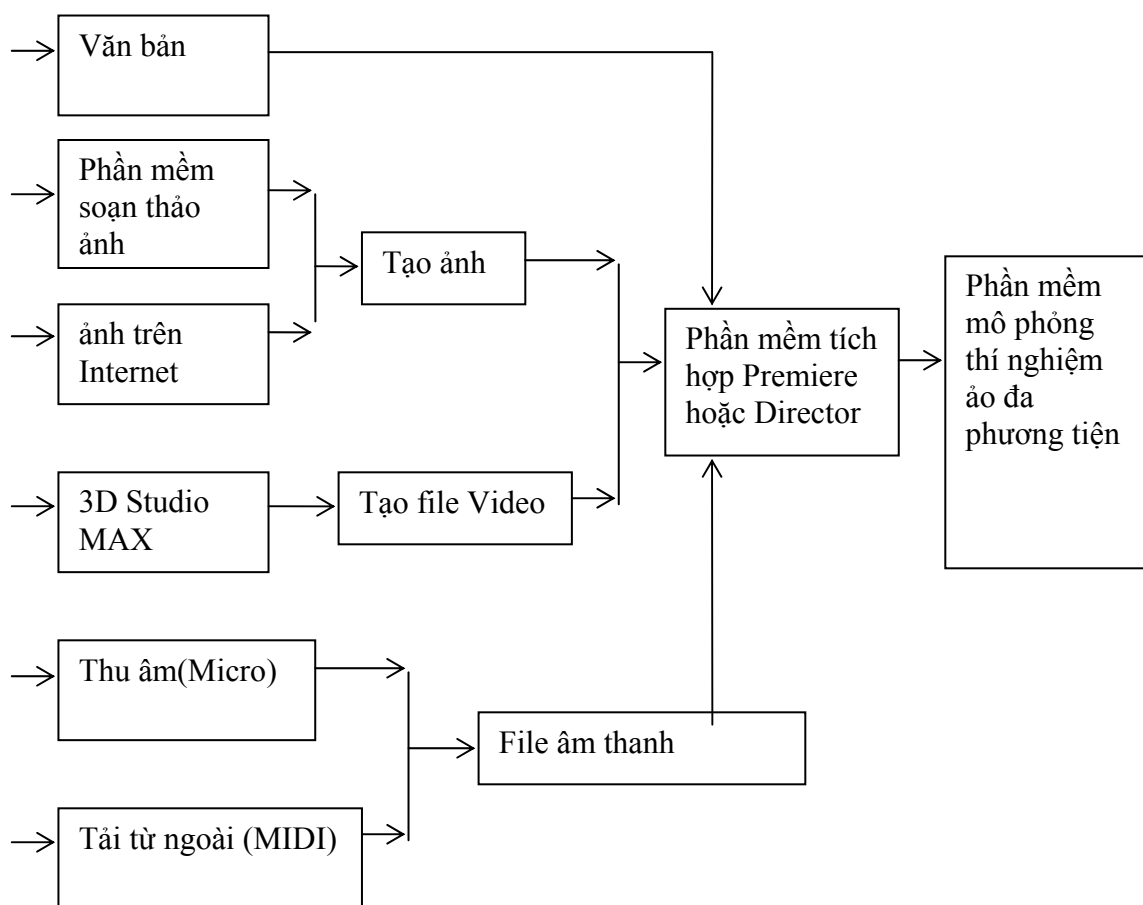
Một số sản phẩm đa phương tiện xây dựng dữ liệu video theo các tham số sau:

- Tên file video: mov + số thứ tự bài + số thứ tự của ảnh trong bài
- Kích thước khung hình video: không quá 800x600 pixels
- Số khung hình trong một giây: 24 fps

- Định dạng file video: *.avi
- Màu sắc: 16 triệu màu
- Thuật toán nén: nén mất thông tin Microsoft DV(NTSC)

1.2. Tích hợp dữ liệu

Tích hợp dữ liệu là bước quan trọng nhất trong quá trình tạo sản phẩm. Sử dụng các dữ liệu đã thu thập được chúng tôi sử dụng phần mềm tích hợp dữ liệu Adobe Premiere hoặc Macromedia Director để tạo thành sản phẩm đa phương tiện.



Quá trình tích hợp dữ liệu để tạo sản phẩm

1.3. Kiểm tra

Kiểm tra là giai đoạn phát hiện lỗi còn lại trong phần mềm. Vì vậy kiểm tra cần được lập kế hoạch và thực hiện một cách cẩn thận và hiệu quả vì chúng ta đều biết không có một đảm bảo nào để chắc chắn không có lỗi dựa trên các kỹ thuật hiện tại kể cả kiểm tra lại nhiều lần. Làm thế nào để phát hiện ra các lỗi còn lại một cách hiệu quả là điều rất quan trọng. Không phải cứ kiểm tra lại nhiều lần là tốt mà phải kiểm tra một cách có hệ thống để phát hiện nhiều nhất những lỗi còn lại.

Chất lượng thiết kế phần kiểm tra phụ thuộc vào kỹ năng và kinh nghiệm của người thiết kế phần kiểm tra. Có một thực tế là do hạn chế về mặt thời gian mà chỉ một số các trường hợp chung nhất mới được kiểm tra mà thôi. Vì thế việc lựa chọn dữ liệu kiểm tra cũng rất quan trọng. Khi thiết kế phần kiểm tra phải chú ý các khía cạnh sau: các chức năng chính, các đối tượng có được xây dựng đúng theo kịch bản không, việc kiểm tra có

ảnh hưởng đến môi trường (phần cứng, phần mềm, mạng), có cần tài liệu hướng dẫn sử dụng không.

Kiểm tra chức năng: sản phẩm đa phương tiện có thể có nhiều module hợp thành và có thể do nhiều người hay nhóm người thực hiện. Khi xây dựng thì chúng ta đi từ tổng thể tới chi tiết nhưng khi thực hiện và kiểm tra lại phải đi từ chi tiết tới toàn bộ. Cần kiểm tra chất lượng của từng module. Kiểm tra tổng thể là bước kiểm tra các liên kết, ghép nối giữa các module trong phần mềm là chính xác, sau đó cần phát hiện những khiếm khuyết khác như sự thích hợp với môi trường, hiệu quả của hệ thống.

Phần kiểm tra quan trọng nhất là người dùng kiểm tra. Có những lỗi là người phát triển không tìm được như sự bất hợp lý trong giao tiếp người máy có thể phát hiện được từ người sử dụng. Một phần mềm đa phương tiện cần phải qua giai đoạn người dùng kiểm tra trước khi trở thành sản phẩm trên thị trường.

2. Kết luận

Từ cách xây dựng qui trình tạo phần mềm mô phỏng thí nghiệm ảo đa phương tiện chúng tôi đã:

1. Tạo ra được một phần mềm mô phỏng thí nghiệm ảo đa phương tiện “Hiện tượng khúc xạ ánh sáng”. Sản phẩm này sẽ được sử dụng trên đĩa CD_ROM như một công cụ hỗ trợ giảng dạy các bài thí nghiệm vật lý lớp 8 phổ thông.
2. Xây dựng kho dữ liệu đa phương tiện có định dạng chuẩn. Các dữ liệu này được quản lý bởi hệ quản trị cơ sở dữ liệu đa phương tiện và sẽ được dùng trong quá trình tạo các sản phẩm khác.

Một số nhận xét rút ra trong quá trình xây dựng quy trình tạo phần mềm mô phỏng thí nghiệm ảo đa phương tiện:

- Loại nội dung cho CD-ROM trên máy tính không phải là định dạng thích hợp cho truyền thụ kiến thức cộng đồng. Chỉ nên dành nó cho việc lưu trữ, tra cứu, học tập cá nhân
- Tùy theo yêu cầu và trình độ người dùng để quyết định mức độ tích hợp đa phương tiện
- Việc ứng dụng kỹ thuật nghe nhìn bằng Digital Video-Audio là điều kiện bắt buộc. Vấn đề này cần hài hòa chất lượng Video với dung lượng bộ nhớ và tốc độ máy (cho đến nay thì MPEG2 có thể coi là tạm đáp ứng yêu cầu)

- Với các phần mềm mô phỏng thí nghiệm ảo đa phương tiện thì việc tích hợp văn bản, âm thanh và video trong môi trường lập trình graphic như Director là đủ.
- Tuy nhiên với các phần mềm hỗ trợ giáo dục phải thiết kế đường dẫn liên kết mạnh và thuận tiện, chức năng ghi nhớ tiện dụng. Các sản phẩm mô phỏng thí nghiệm ảo hiện nay trên thị trường là chưa tối ưu. Có thể phải sử dụng kỹ thuật khác hơn là ***tạo mối liên kết giữa các trang màn hình***.
- Các chương trình giáo dục trong nhà trường đòi hỏi tích hợp Multimedia cao nhất và độ tương tác cũng phải cao. Như vậy các công cụ hiện có cần phải kết hợp với nhau. Mặt khác người thiết kế cần phải tự lập trình bằng các ngôn ngữ cao như C, ngay từ đầu thì mới có thể đáp ứng các yêu cầu cao, như các mô phỏng ảo chẳng hạn, không thể dựa vào công cụ sẵn có được.
- Cần có sự hợp tác rộng rãi và nỗ lực hơn nữa giữa các nhà sư phạm và người thiết kế, lập trình thì mới có thể tạo ra các phần mềm nội dung tốt, kỹ thuật cao cho người dùng.

3. Phụ lục - Xây dựng phần mềm Hiện tượng khúc xạ ánh sáng

Từ nhiều năm nay, trong quá trình cải cách và phát triển giáo dục ở nước ta, mục tiêu, chương trình, nội dung giáo dục đã được thay đổi nhiều lần cho phù hợp với yêu cầu phát triển kinh tế-xã hội và đã đạt được những tiến bộ đáng kể. Trong khi đó, những thay đổi trong phương pháp giảng dạy nói chung còn quá ít và quá chậm so với nhu cầu của người học, nên đã hạn chế đáng kể đến chất lượng đào tạo học sinh. Do đó cần phải đổi mới phương pháp dạy học ở tất cả các cấp học, bậc học, cần áp dụng những phương pháp giáo dục hiện đại để bồi dưỡng cho học sinh năng lực tư duy sáng tạo, năng lực giải quyết vấn đề.

Việc đổi mới phương pháp giáo dục nhằm phát huy tính tích cực, chủ động, năng lực tự học, tự nghiên cứu, năng lực đặt vấn đề và giải quyết vấn đề ngay trong quá trình học tập của người học.

Cho tới ngày nay, đại đa số giáo viên khi giảng dạy cũng chỉ sử dụng những phương tiện truyền thông nghe nhìn thông thường như sơ đồ, bảng biểu, bản đồ,... và các thao tác truyền đạt khi giảng dạy còn rất thủ công. Điều đó tuy có một vài ưu điểm như ít tốn kém về đầu tư vào cơ sở vật chất và không phải huấn luyện lại giáo viên... thế nhưng nhược điểm cũng bộc lộ căn bản là phụ thuộc hoàn toàn vào trình độ và trạng thái tâm

sinh lý lên lớp của giáo viên, rất hạn chế về các hình thức thể hiện kiến thức và không minh họa các kiến thức động.

Với phần mềm mô phỏng thí nghiệm ảo vật lý “Hiện tượng khúc xạ ánh sáng”, người học là đã có thể tiếp thu được tất cả những gì mà một giáo viên có thể truyền đạt trên lớp, thêm vào đây họ còn được xem những hình ảnh minh họa nội dung thí nghiệm (động hoặc tĩnh). Chương trình có chức năng cho người sử dụng biết họ đã hiểu bài học chưa, nếu chưa thì như thế nào mới là đúng. Họ có thể học đi, học lại một bài bất kì cho đến khi trả lời được hầu hết các câu hỏi nêu ra. Chương trình cung cấp hiện tượng thí nghiệm và cả hình minh họa nội dung bài thí nghiệm. Những đặc điểm này sẽ khắc phục được sự bất cân đối về học lực của người sử dụng, làm bài thí nghiệm đơn giản và dễ hiểu hơn.

Chương trình có những tính năng hỗ trợ giảng dạy như xem, nghe nội dung bài học, trắc nghiệm kiến thức, tra cứu từ ngữ, quan sát hình ảnh.

Quá trình xây dựng một chương trình dạy học tất nhiên phải tuân thủ những nguyên tắc sư phạm. Yếu tố sư phạm được thể hiện ở chỗ từ yêu cầu về nội dung của bài thí nghiệm và rồi trong mỗi bài lại được chia nhỏ hơn thành những đơn vị tri thức. Mục tiêu mỗi hiện tượng thí nghiệm là đưa đến người học những đơn vị tri thức đó.

Hiện tượng khúc xạ ánh sáng là kiến thức vật lý cơ bản trong sách giáo khoa lớp 8 và lớp 12 phổ thông. Trong phần kiến thức này yêu cầu giáo viên hướng dẫn thí nghiệm hiện tượng khúc xạ ánh sáng phải mô tả lại các tượng khúc xạ mà học sinh có thể nhìn thấy bằng trực quan (chiếc đĩa cắm vào cốc huỷ tinh-chiếc đĩa thẳng nhưng khi đổ nước vào cốc, chiếc đĩa hình như bị gãy khúc) sau đó tiến hành thí nghiệm, mô hình hóa thí nghiệm đó và đưa ra các kết luận về hiện tượng khúc xạ ánh sáng. Học sinh yêu cầu phải nhìn rõ các hiện tượng, hiểu thí nghiệm, có thể thao tác lại và đưa ra những nhận xét cá nhân và hoặc liên hệ với các hiện tượng liên quan khác (hiện tượng phản xạ ánh sáng, sự truyền thẳng của ánh sáng.)

Phần mềm “Hiện tượng khúc xạ ánh sáng” có chức năng chính là hỗ trợ giáo viên mô hình hóa các thí nghiệm, hiện tượng. Nó được sử dụng với mục đích chính là công cụ để qua đó giáo viên có thể thực hiện các tương tác, thí nghiệm nằm trong phần nội dung hiện tượng khúc xạ ánh sáng trong chương trình giáo dục vật lý lớp 8. Ngoài ra sản phẩm này hỗ trợ học sinh tìm hiểu sâu hơn về thí nghiệm hiện tượng khúc xạ ánh sáng, học sinh có thể học tại nhà hoặc bất kỳ đâu tại bất cứ lúc nào. Phần mềm này đặc biệt có ý nghĩa quan trọng trong điều kiện các nhà trường phổ thông ở Việt Nam hiện nay thường xuyên thiếu các trang thiết bị thí nghiệm và các trường phổ thông ở vùng sâu vùng xa.

Chính vì những lý do trên, sau khi tìm hiểu kỹ quy trình xây dựng phần mềm mô phỏng thí nghiệm ảo đa phương tiện chúng tôi đã xây dựng phần mềm “Hiện tượng khúc xạ ánh sáng” theo kế hoạch và nội dung thực hiện như sau:

3.1. Kế hoạch xây dựng phần mềm

Theo kế hoạch phần mềm hiện tượng khúc xạ ánh sáng được xây dựng trong 5 tháng và kết quả cần thực hiện là tạo một sản phẩm đóng gói trên đĩa CD và xây dựng tập đối tượng dữ liệu vật lý thuộc nội dung hiện tượng khúc xạ ánh sáng.

Để xây dựng được phần mềm này nhóm chúng tôi cần phải chia thành các nhóm nhỏ sau:

- Quản lý
- Đạo diễn
- Họa sỹ thiết kế
- Nhóm xây dựng đối tượng dữ liệu và mô phỏng hiện tượng
- Nhóm lập trình tạo tương tác giữa các đối tượng dữ liệu và đóng gói sản phẩm

Mỗi nhóm có trách nhiệm hoàn thành các công việc của mình như trong kịch bản và kế hoạch đã đề ra. Cụ thể chúng tôi có tiến độ thực hiện như sau:

STT	Nội dung các bước tiến hành	Thời gian		Nhóm thực hiện
		Bắt đầu	Kết thúc	
1.	Lập và xét duyệt đề cương		31/11/02	Quản lý
2.	Thu thập dữ liệu	01/12/02	15/01/03	Nhóm xây dựng và lập trình
3.	Đọc, thảo luận và thống nhất kịch bản	16/01/03	15/02/03	Đạo diễn, họa sỹ thiết kế và nhóm xây dựng và lập trình
4.	Xây dựng dữ liệu về Video, Audio dựa trên kịch bản và hoàn thành sản phẩm	16/02/03	16/03/03	Họa sỹ thiết kế, nhóm xây dựng và lập trình
5.	Kiểm tra sản phẩm	16/03/03	25/03/03	Người dùng, nhóm xây dựng và lập trình
6.	Hướng dẫn và triển khai hệ thống	26/03/03	31/03/03	Nhóm xây dựng và lập trình

7.	Nộp kết quả, báo cáo		31/03/03	Quản lý
----	----------------------	--	----------	---------

3.2. Nội dung thực hiện

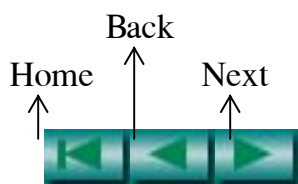
3.2.1 Thống nhất kịch bản

Sau khi nhận được kịch bản từ Viện Giáo Dục, chúng tôi tìm hiểu và thống nhất kịch bản như sau:

Màn hình thể hiện có kích thước 800x600. Hình thức sử dụng chương trình là các mối liên kết giữa trang này với trang kia. Sau màn hình giới thiệu thì người dùng sẽ nhìn thấy trang đầu tiên mô tả hiện tượng. Mỗi thí nghiệm đều có phần ghi nhớ và kết luận giúp người học củng cố kiến thức về hiện tượng hay thí nghiệm đó. Mỗi hiện tượng lại bao gồm các phần nhỏ thể hiện bởi các đề mục chỉ rõ nội dung, mục đích thí nghiệm, đối tượng. Qua đó giúp cho người học nhanh chóng thâm nhập và hiểu được mình đang ở đâu và làm gì trong quá trình quan sát thí nghiệm. Cuối mỗi bài thí nghiệm đều có phần mô hình hóa kết quả thí nghiệm bằng file chạy video.

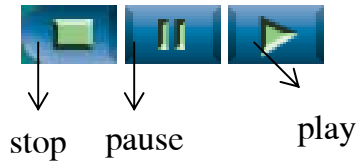
Để quản lý thao tác của người dùng thì chương trình xây dựng một số hiệu ứng xảy ra khi di chuột qua đối tượng hoặc bấm chuột vào đối tượng điều khiển. Cụ thể là khi trở chuột vào một đoạn văn bản hoặc biểu tượng mang tính liên kết thì đoạn văn bản sẽ đổi màu và hình dạng con chuột cũng thay đổi. Khi trở chuột vào nút liên kết có một đoạn chú thích hiện ra phía trên chuột cho người dùng biết đấy là nút gì đồng thời hình dạng của nút sẽ thay đổi để người dùng chắc chắn rằng mình đang chuẩn bị tác động vào nút đó. Khi bấm chuột vào biểu tượng liên kết người dùng sẽ thấy biểu tượng thay đổi và có hình bàn tay liên kết.

Những tính năng của chương trình đã được thực hiện:



- Nút Menu: cho người dùng sẽ quay về trang chủ dù đang đứng ở bất kì cảnh nào. Tại đó chứa tất cả các thí nghiệm trong “hiện tượng khúc xạ ánh sáng”. Trong trang chủ ta nhấn chuột vào biểu tượng tiếp theo để quan sát hiện tượng đầu tiên của thí nghiệm

- Nút Back: khi người dùng nhấn chuột vào nút này thì chương trình sẽ quay về thí nghiệm trước đó. Do có sự liên kết này mà người quay trở về trang trước xem xét, đánh giá lại các thí nghiệm đã quan sát trước đó.
- Nút Next: khi người dùng nhấn chuột vào đây thì chương trình sẽ chuyển đến thí nghiệm tiếp theo.



Đây là ba nút điều khiển có tác dụng khi nghe giảng. Chúng tương tự như các nút điều khiển ở trong cassette. Nhấn chuột vào nút thứ ba, tức là nút *play*, chương trình sẽ đọc nhận xét và kết luận về hiện tượng đang xảy ra. Nhấn chuột vào nút thứ hai, tức là nút *pause*, việc nghe bài giảng sẽ bị tạm dừng và để nghe tiếp thì lại nhấn chuột một lần nữa vào nút *pause*. Nút đầu tiên- nút *stop*, kích chuột vào nút này cũng giống như việc tắt cassette.

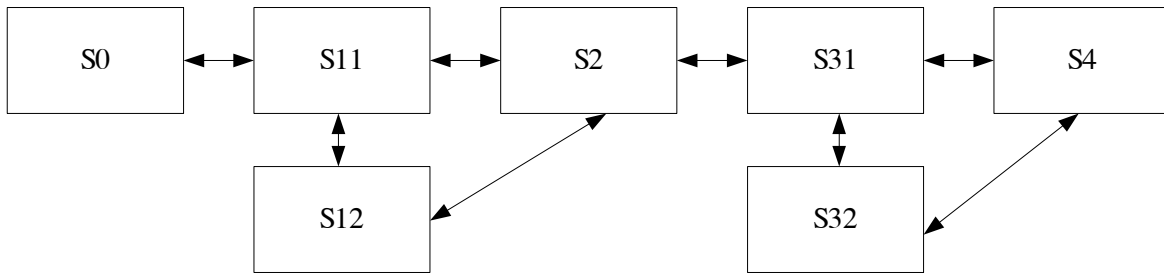
Trong mỗi phần thí nghiệm, có một số dòng chữ có tác dụng liên kết. Khi nhấn chuột vào đây thì sẽ có hình ảnh hiện lên để minh họa hoặc giải thích cho các thuật ngữ sử dụng trong thí nghiệm. Đặc biệt đối với người dùng mới họ có thể xem hướng dẫn cách sử dụng chương trình ở bất kỳ màn hình nào bằng cách kích chuột vào biểu tượng “hướng dẫn sử dụng” ở góc phải trên cùng màn hình. Khi người dùng cần đọc mô tả chi tiết bằng văn bản hiện tượng thí nghiệm thì họ có thể bấm chuột vào biểu tượng ghi nhớ nhận xét góc bên phải mỗi thí nghiệm.

Phía dưới góc trái màn hình còn có nút *Sound* cho người dùng bật/tắt nhạc và điều chỉnh cường độ âm thanh.

Phía dưới ở giữa màn hình có nút *Exit* cho phép người dùng thoát khỏi chương trình dù ở bất kỳ trang màn hình nào.

Trong phần tương tác với đối tượng, người dùng có thể dùng chuột điều khiển, di chuyển các đối tượng để quan sát hiện tượng dưới nhiều góc độ khác nhau.

Sơ đồ tổng thể của chương trình:



Sơ đồ tổng thể của chương trình

Giải thích kí hiệu:

S#0: Màn hình chủ sự khúc xạ ánh sáng

S#1i: Pha 1- Nêu vấn đề

S#2: Pha 2- Hiện tượng khúc xạ ánh sáng

S#3i: Pha 3- Khảo sát hiện tượng khúc xạ ánh sáng

S#4: Pha 4- Sự đổi hướng phụ thuộc tính chất của 2 môi trường

Giải thích sơ đồ:

Từ cảnh S#0, chương trình sẽ tự động chuyển sang cảnh S#1. Trong cảnh S#1 có hiện tượng 1 và hiện tượng 2. Từ cảnh S#1 có các mối liên kết đến các cảnh S#2, từ cảnh S#2 có mối liên kết tới cảnh S#3. Trong cảnh S#3 có hai thí nghiệm là thí nghiệm 2 và thí nghiệm 3. Từ cảnh S#3 có mối liên kết tới cảnh S#4. Từ các cảnh S#i (i=1..4) ta có thể trực tiếp quay về cảnh S#0.

a. Danh sách cảnh

STT	Tên cảnh	Tình huống
S#0	Giới thiệu	Màn hình giới thiệu hiện tượng khúc xạ ánh sáng
S#1.1	Pha 1- Hiện tượng 1	Màn hình giới thiệu hiện tượng chiếc đĩa cắm vào cốc thuỷ tinh khi đổ nước vào quan sát thì thấy chiếc đĩa hình như bị gãy. Khi nhấn chuột vào các biểu tượng liên kết thì chương trình sẽ chuyển sang hiện tượng 2 hoặc sang thí nghiệm 1 (S#2)

S#1.2	Pha 1- Hiện tượng 2	Màn hình giới thiệu hiện tượng nhìn đồng xu dưới đáy chậu nước, quan sát hiện tượng thấy hình như vật ở gần mặt nước hơn. Khi nhấn chuột vào các biểu tượng liên kết thì chương trình sẽ quay về hiện tượng 1 hoặc chuyển sang thí nghiệm 1 (S#2)
S#2.	Pha 2 - Thí nghiệm 1- Chiếu tia sáng từ không khí vào nước	Màn hình mô tả thí nghiệm chiếu tia sáng từ không khí vào nước. Khi kích chuột vào các biểu tượng thì các đối tượng cần cho thí nghiệm sẽ được hiển thị, tiến hành thí nghiệm, dùng chuột di chuyển đối tượng để quan sát hiện tượng thay đổi theo. Kích chuột vào biểu tượng ghi nhớ để xem lời nhận xét. Kích chuột vào nút Next để xem mô hình hóa thí nghiệm 1.
S#3.1	Pha 3- Thí nghiệm 2 – Cho ánh sáng đi từ không khí vào nước	Màn hình mô tả thí nghiệm chiếu một chùm sáng từ không khí vào nước, đến mặt nước chùm sáng đổi hướng đi vào nước. Khi kích chuột vào các biểu tượng thì các đối tượng cần cho thí nghiệm sẽ được hiển thị, tiến hành thí nghiệm, dùng chuột di chuyển đối tượng để quan sát hiện tượng thay đổi theo. Kích chuột vào biểu tượng ghi nhớ để xem lời nhận xét. Kích chuột vào nút Next để xem mô hình hóa thí nghiệm 2.
S#3.2	Pha 3- Thí nghiệm 3 – Cho ánh sáng đi từ nước vào không khí	Màn hình mô tả thí nghiệm chiếu một chùm sáng đi từ nước ra ngoài không khí, đến mặt nước chùm sáng đổi hướng đi ra ngoài không khí. Khi kích chuột vào các biểu tượng thì các đối tượng cần cho thí nghiệm sẽ được hiển thị, tiến hành thí nghiệm, dùng chuột di chuyển đối tượng để quan sát hiện tượng thay đổi theo. Kích chuột vào biểu tượng ghi nhớ để xem lời nhận xét. Kích chuột vào nút Next để xem mô hình hóa thí nghiệm 2.
S#4	Pha 4- Thí nghiệm 4 – Sự	Màn hình mô tả thí nghiệm khi thay đổi chiết suất giữa 2 môi trường. Khi kích chuột vào các biểu tượng thì

	đổi hướng phụ thuộc tính chất của 2 môi trường	các đối tượng cần cho thí nghiệm sẽ được hiển thị, tiến hành thí nghiệm, dùng chuột di chuyển đối tượng để quan sát hiện tượng thay đổi theo. Kích chuột vào biểu tượng ghi nhớ để xem lời nhận xét. Kích chuột vào nút Next để xem mô hình hóa thí nghiệm 4.
--	--	---

b. Danh sách sự kiện

STT	Tên sự kiện	Đích	Kích hoạt	Sự kiện
S#0	Chuyển đến hiện tượng hoặc thí nghiệm đầu tiên	<i>Text member</i>	- Nhấn chuột. - Thay đổi hình dạng chuột.	Khi kích chuột vào các xâu văn bản thì chương trình sẽ chuyển sang cảnh tương ứng.
		<i>Nút Next</i>	- Nhấn chuột. - Thay đổi hình dạng chuột. - Dòng chú thích gắn kèm với chuột.	Khi nhấn chuột vào nút này chương trình sẽ chuyển sang cảnh S#1.i (i=1..4)
	Thoát khỏi chương trình	<i>Nút Exit</i>	- Nhấn chuột	Kích chuột vào nút Exit để thoát khỏi chương trình
S#1.1 (S#1.2)	Chuyển về màn hình chủ.	<i>Nút Menu</i>	- Nhấn chuột	Chương trình sẽ chuyển về trang màn hình chủ nếu nhấn nút <i>menu</i> .

	Chuyển đến hiện tượng 2.	Nút <i>Hiện tượng 2(1)</i>	- Nhấn chuột	Chương trình sẽ chuyển đến màn hình hiện tượng 2 nếu nhấn vào nút <i>Hiện tượng 2</i> .
	Thoát khỏi chương trình	Nút <i>Exit</i>	- Nhấn chuột	Kích chuột vào nút Exit để thoát khỏi chương trình
S#2 S#3 S#4	Hiện đối tượng thí nghiệm.	Các nút biểu tượng liên kết.	- Nhấn chuột	Khi bấm chuột vào nhóm từ liên kết, người học sẽ được xem ảnh minh họa.
	Di chuyển đối tượng	Các hình ảnh đối tượng	- Di chuyển chuột	Dùng chuột di chuyển đối tượng tới các vị trí và đối tượng khác nhau trong khi quan sát hiện tượng đang xảy ra trong thí nghiệm
	Chuyển về màn hình chủ.	Nút <i>Menu</i>	- Nhấn chuột	Chương trình sẽ chuyển về trang màn hình chủ nếu nhấn nút <i>menu</i>
	Chuyển về thí nghiệm trước.	Nút <i>Back</i>	- Nhấn chuột	Khi nhấn chuột vào nút <i>Back</i> , chương trình sẽ chuyển đến cảnh tiếp theo.
	Thoát khỏi chương trình	Nút <i>Exit</i>	- Nhấn chuột	Kích chuột vào nút Exit để thoát khỏi chương trình

	Nghe nhận xét rút ra từ thí nghiệm.	Nút <i>Play</i> .	- Nhấn chuột	Khi người học nhấn chuột vào nút <i>play</i> tương đương với việc bật băng để nghe kết luận và nhận xét được rút ra từ thí nghiệm.
	Tạm dừng.	Nút <i>Pause</i> .	- Nhấn chuột	Khi kích chuột vào nút <i>pause</i> thì việc nghe sẽ bị tạm dừng và khi nhấn vào nút <i>play</i> thì người dùng sẽ được nghe tiếp sau đoạn tạm dừng lúc trước.
	Tắt âm thanh	Nút <i>stop</i>	- Nhấn chuột	Tác dụng chuột vào nút <i>stop</i> cũng tương tự như nút <i>pause</i> , chỉ khác là khi ta tác động vào nút này và sau đó tác động vào nút <i>play</i> thì ta sẽ nghe lại bài hội thoại từ đầu.

c. Giao diện người dùng

S#0: Màn hình chung

Tên sản phẩm: Sự khúc xạ ánh sáng				
Mô tả nội dung hiện tượng khúc xạ ánh sáng sẽ được trình bày trong phần tới				
Âm thanh	Thoát	Trước	Trang chủ	Sau

S#1: Pha 1- Hiện tượng khúc xạ ánh sáng

Hiện tượng khúc xạ ánh sáng				
Mô hình hoá hiện tượng (Video)			Quan sát hiện tượng 1 (chiếc đũa cắm vào cốc thuỷ tinh có nước)	
Dừng	Tạm dừng	Chạy		
Âm thanh	Thoát	Trước	Trang chủ	Sau

S#2: Pha 2- Thí nghiệm 1- Chiếu tia sáng từ không khí vào nước

Chiếu tia sáng từ không khí vào nước		
Nhận xét	Mô phỏng hiện tượng, mô hình hoá thí nghiệm	Hướng dẫn

Các đối tượng thí nghiệm				
				Mô hình hoá thí nghiệm
Âm thanh	Thoát	Trước	Trang chủ	Sau

S#3: Pha 3- Khảo sát hiện tượng khúc xạ

Khảo sát hiện tượng khúc xạ				
Nhận xét	Mô phỏng hiện tượng chiếu chùm sáng từ không khí vào nước và cho ánh sáng đi từ nước ra ngoài không khí, mô hình hoá thí nghiệm			Hướng dẫn
Các đối tượng thí nghiệm				Mô hình hoá thí nghiệm
Âm thanh	Thoát	Trước	Trang chủ	Sau

S#4: Pha 4- Quan hệ góc khúc xạ và chiết suất của 2 môi trường

Quan hệ góc khúc xạ và chiết suất của 2 môi trường		
Nhận xét	Mô phỏng hiện tượng chiếu chùm sáng từ không khí vào các môi trường khác nhau: nước, dầu, thủy tinh	Hướng dẫn
Đối tượng thí nghiệm		

Các giá trị thí nghiệm				Mô hình hoá thí nghiệm
Âm thanh	Thoát	Trước	Trang chủ	Sau

Màn hình mô hình hoá thí nghiệm

Tên thí nghiệm, hiện tượng trong S#i (i=1..4)				
Mô hình hoá thí nghiệm bằng file chạy video				
	Dừng	Tạm dừng	Chạy	Thanh trượt
Âm thanh	Thoát	Trước	Trang chủ	Sau

Màn hình nhận xét, kết luận rút ra từ thí nghiệm:

Tên thí nghiệm, hiện tượng trong cảnh S#i (i=1..4)			
- Văn bản ghi nhớ những nhận xét, kết luận rút ra qua thí nghiệm đang mô tả - Một số đề xuất, gợi ý, liên tưởng tới các hiện tượng, thí nghiệm liên quan (nếu có)		Hình ảnh tĩnh minh hoạ chi tiết hơn về thí nghiệm, hiện tượng trong cảnh S#i (i=1..4)	
Âm thanh	Thoát	Trang chủ	Trước

Màn hình hướng dẫn, trợ giúp:

Tên sản phẩm: Sự khúc xạ ánh sáng			
Văn bản hướng dẫn sử dụng phần mềm Địa chỉ liên lạc hỗ trợ giải đáp kỹ thuật, hướng dẫn sử dụng			
Thoát	Trước	Trang chủ	Sau

1.1.1 Cài đặt phần mềm

Trong quá trình xây dựng sản phẩm chúng tôi đã sử dụng các phần mềm sau:

1. **Microsoft Word:** tạo văn bản, định dạng phong chữ, kiểu text
2. **PaintShopPro7** và **PhotoShop6:** tạo hình ảnh 2 chiều
3. **3D Studio Max:** tạo hình ảnh 3 chiều, mô hình hoá các đối tượng 3 chiều
4. **Snapit 6:** thu âm thanh
5. **Adobe Premiere 6:** chỉnh sửa các đoạn video và âm thanh
6. **Hệ quản trị cơ sở dữ liệu đa phương tiện:** quản lý toàn bộ các file dữ liệu
7. **Macromedia Director 8:** lập trình tạo tương tác giữa các đối tượng, ghép nối chương trình.

1.1.2 Yêu cầu phần cứng

- Máy tính
- CD - ROOM
- Loa và bìa âm thanh
- Máy ảnh số
- Máy quay video số
- Thiết bị chuyển đổi tín hiệu DV300 và card 1394 chuyển đổi tín hiệu từ máy quay video qua máy tính.

- CD Rewriteable

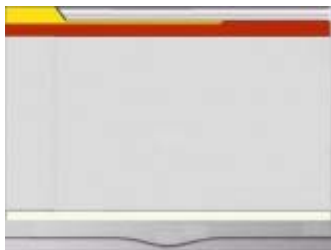
1.1.3 Xây dựng và quản lý kho dữ liệu đồ họa cơ bản

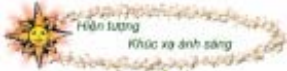
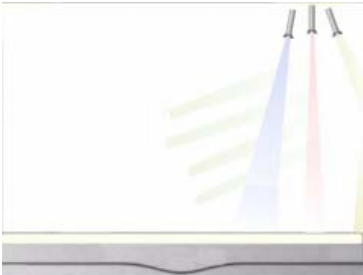
a. Văn bản

STT	Tên cảnh	Văn bản
1.	S#0: Màn hình chung	- Sự khúc xạ ánh sáng - Liệu có phải ánh sáng luôn truyền theo đường thẳng?
2.	S#1.1 Hiện tượng 1	- Hãy quan sát cái đĩa đặt trong một cốc thủy tinh khi không có nước như hình bên. - Điều gì sẽ xảy ra khi ta đổ nước vào cốc?
3.	S#1.1. Mô hình hóa hiện tượng 1	- Nhìn cốc nước, ta thấy: - Hình như đĩa bị gãy tại mặt nước và đáy dường như được nâng lên cao. - Phần đĩa ở dưới nước như to hơn phần đĩa trên mặt nước
4.	S#1.2 Hiện tượng 2	- Hãy quan sát đặt trong một cốc thủy tinh. Quan sát chiếc vòng bạc
5.	S#3 Khảo sát hiện tượng khúc xạ	- Hiện tượng ánh sáng khi truyền từ môi trường trong suốt này sang môi trường trong suốt khác bị gãy khúc tại mặt phân cách giữa hai môi trường được gọi là hiện tượng khúc xạ ánh sáng. - Tia khúc xạ luôn luôn nằm trong mặt phẳng tới. - Khi ánh sáng truyền từ không khí vào nước, góc tới lớn hơn góc khúc xạ - Khi ánh sáng truyền từ nước vào không khí, góc tới nhỏ hơn góc khúc xạ. - Mô hình giải thích hai hiện tượng nêu ra ban đầu: Các tia sáng phát ra từ một nguồn sáng S ở trong nước đến mặt nước bị khúc xạ, do đó chùm tia sáng đến mắt người quan sát làm cho người quan sát có cảm giác là chúng xuất phát từ S'. S' ở gần mặt nước hơn.

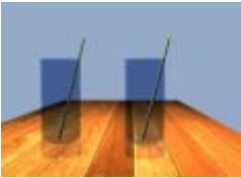
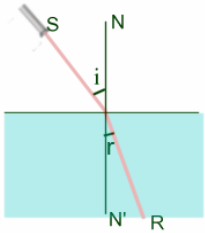
		- Để mắt ở người không khí nhìn một vật ở trong nước, do hiện tượng khúc xạ ánh sáng, người ta không thể nhìn được chính vật ấy mà chỉ nhìn thấy ảnh của nó ở gần mặt nước hơn. Chiếc đĩa bị “gãy khúc” khi cắm vào trong cốc nước cũng được giải thích tương tự. Mắt ta không nhìn thấy chính phần chiếc đĩa ở trong nước mà chỉ nhìn thấy ảnh của phần đó, do đó ta có cảm giác chiếc đĩa bị “gãy khúc”
6.	S#4 Quan hệ giữa góc khúc xạ và chiết suất của 2 môi trường	- Ta đã quan sát hiện tượng ánh sáng truyền giữa hai môi trường không khí và nước, vậy ánh sáng sẽ truyền như thế nào khi đi từ không khí vào các môi trường khác? Liệu môi trường truyền có ảnh hưởng gì đến độ lớn của góc khúc xạ? - Tiến hành thí nghiệm sau và rút ra câu trả lời! - Khi ánh sáng đi từ không khí vào các môi trường trong suốt rắn, lỏng khác nhau thì: (i) góc tới lớn hơn góc khúc xạ, (ii) khi góc tới tăng (giảm) thì góc khúc xạ cũng tăng (giảm) tương ứng, (iii) cùng một góc tới, với các môi trường khác nhau thì góc khúc xạ khác nhau, (iv) khi tia tới vuông góc với mặt phân cách giữa hai môi trường thì truyền thẳng, tia sáng không bị khúc xạ

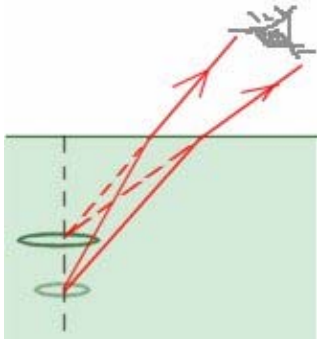
b. Hình ảnh

STT	Tên file	Mô tả	Cảnh sử dụng	Chỉnh sửa /Mục đích sử dụng
1.	Mygd3.jpg		S#0 S#1 S#2 S#3	ảnh nền

			S#4	
2.	Logo.gif		S#0	ảnh logo
3.	Anhnen.jpg		S#0	ảnh nền
4.	Trogiup.jpg		S#0 S#1 S#2 S#3 S#4	ảnh biểu tượng liên kết hướng dẫn trợ giúp
5.	Qsht.jpg		S#0 S#1	Đổi màu/biểu tượng tạo liên kết tới trang khác
6.	Loa.png		S#0 S#1 S#2 S#3 S#4	Biểu tượng bật tắt loa
7.	Thoat.png		S#0 S#1 S#2 S#3	Biểu tượng thoát khỏi chương trình

			S#4	
8.	Thanh.jpg		S#0 S#1 S#2 S#3 S#4	Biểu tượng thanh trượt điều khiển âm thanh
9.	Conchay.jpg		S#0 S#1 S#2 S#3 S#4	Biểu tượng thanh trượt điều khiển âm thanh
10.	Home.gif		S#1 S#2 S#3 S#4	Biểu tượng liên kết tới màn hình chủ, khi di chuột tới thì thay đổi màu sắc
11.	Back.gif		S#1 S#2 S#3 S#4	Biểu tượng liên kết tới màn hình sau, khi di chuột tới thì thay đổi màu sắc
12.	Next.gif		S#0 S#1 S#2 S#3	Biểu tượng liên kết tới màn hình trước đó, khi di chuột tới thì thay đổi màu sắc

13.	Haicoc.jpg		S#1	Hình ảnh minh họa
14.	Tamgo.jpg		S#2 S#3 S#4	- Biểu tượng kích hoạt - Hình ảnh minh họa
15.	Chaunuoc.jpg		S#2 S#3 S#4	- Biểu tượng kích hoạt - Hình ảnh minh họa
16.	Nhanxet.jpg		S#2 S#3 S#4	- Biểu tượng kích hoạt - Hình ảnh minh họa
17.	Batden.jpg		S#2 S#3 S#4	- Biểu tượng kích hoạt - Hình ảnh minh họa
18.	Den.jpg		S#2 S#3 S#4	- Biểu tượng kích hoạt - Hình ảnh minh họa
19.	Htkx.jpg		S#2 S#3	Hình ảnh minh họa

20.	Kx.jpg		S#3	Hình ảnh mô tả hiện tượng
21.	Dodo.jpg		S#4	- Biểu tượng kích hoạt - Hình ảnh minh họa
22.	Hangso.png		S#4	- Biểu tượng tương tác, kích hoạt
23.	Giaithich.png		S#3	- Hình ảnh minh họa

c. Âm thanh

STT	Màn hình	Kích hoạt	Lời thoại
1.	S#0- Màn hình chung	Bật loa	- Trong bài học này chúng ta cùng tìm hiểu hiện tượng khúc xạ ánh sáng
2.	S#1.1- Quan sát hiện tượng khúc xạ ánh sáng 1	Bật loa	- Hiện tượng 1: Chiếc đũa cắm vào cốc thủy tinh, chiếc đũa thẳng. Khi đổ nước vào cốc, chiếc đũa hình như bị gãy - Có thể giải thích hiện tượng này

			như thế nào?
3.	S#12- Quan sát hiện tượng khúc xạ ánh sáng 2	Bật loa	- Hiện tượng 2: Nhìn chiếc vòng bạc dưới đáy cốc nước, hình như nó ở gần mặt nước hơn - Có thể giải thích hiện tượng này như thế nào?
4.	S#2- Thí nghiệm 1- Hiện tượng khúc xạ ánh sáng	Bật loa	- Thí nghiệm 1- Chiếu tia sáng từ không khí vào nước
		Nguồn sáng	- Chọn nguồn sáng
		Bình thuỷ tinh	- Đặt bình thuỷ tinh trong suốt
		Tấm gỗ	- Đặt tấm gỗ để hứng tia sáng
		Đổ nước vào bình	- Đổ dần dần nước vào bình thuỷ tinh
		Bật loa	- Chiếu chùm sáng vào mặt nước, đến mặt nước chùm sáng đột ngột đổi hướng đi vào trong nước
	Màn hình mô hình hoá TN	Bật loa	- Hình vẽ biểu diễn đường đi của tia sáng từ không khí vào nước - SI là tia tới - IR là tia khúc xạ - Mặt phẳng (SI,IN) là mặt phẳng tới
	S#31- Thí nghiệm 2- Khảo sát hiện tượng khúc xạ	Bật loa	- Thí nghiệm 2-Chiếu chùm sáng từ không khí vào mặt nước, đến mặt nước chùm sáng đổi hướng đi vào nước
		Nguồn sáng	- Chọn nguồn sáng

5.		Bình thuỷ tinh	- Đặt bình thuỷ tinh trong suốt
		Tấm gỗ	- Đặt tấm gỗ để hứng tia sáng
		Đổ nước vào bình	- Đổ dần dần nước vào bình thuỷ tinh
		Dùng chuột di chuyển vị trí tia tới	- Thay đổi vị trí chùm tia tới (cho góc tới thay đổi), góc khúc xạ thay đổi theo
	Mô hình hóa thí nghiệm 2	Bật loa	- Hình ảnh mô tả hiện tượng chiếu chùm sáng từ không khí vào nước, đến mặt nước chùm sáng đổi hướng đi vào nước - Qua thí nghiệm trên có kết luận: góc khúc xạ phụ thuộc góc tới, khi góc tới thay đổi thì góc khúc xạ cũng thay đổi. - Trong trường hợp ánh sáng đi từ không khí vào nước thì góc khúc xạ nhỏ hơn góc tới
	S#3.2- Thí nghiệm 3- ánh	Bật loa	- Thí nghiệm 3-Cho ánh sáng đi từ nước ra ngoài không khí

6.	sáng từ nước ra ngoài không khí	Nguồn sáng	- Đặt nguồn sáng vào trong bình nước (đáy bình)
		Bình thuỷ tinh	- Đặt bình thuỷ tinh trong suốt
		Tấm gỗ	- Đặt tấm gỗ vuông góc với mặt nước
		Đổ nước vào bình	- Đổ dần dần nước vào bình thuỷ tinh
		Bật nguồn sáng	- Bật nguồn sáng, chùm sáng đi từ trong nước, tới mặt phân cách với không khí thì đột nhiên bị đổi hướng ngả về phía mặt nước
		Dùng chuột điều khiển tấm gỗ	- Điều chỉnh tấm gỗ để chùm sáng nằm là là trên mặt nước
7.	Mô hình hóa thí nghiệm 3	Dùng chuột di chuyển vị trí tia tới	- Thay đổi vị trí chùm tia tới (cho góc tới thay đổi), góc khúc xạ thay đổi theo

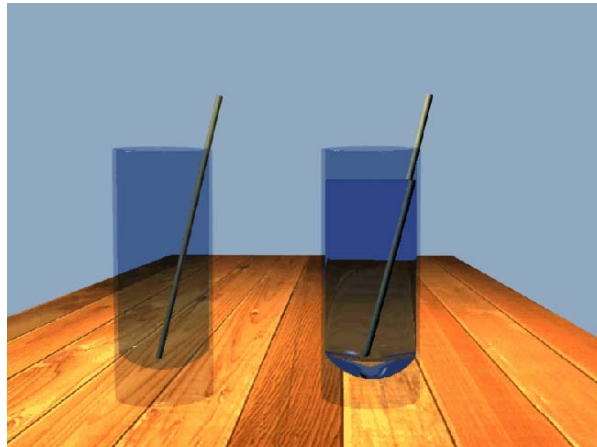
		Bật loa	- Hình ảnh mô tả hiện tượng chiếu chùm sáng từ nước vào không khí, tới mặt phân cách với không khí thì đột nhiên bị đổi hướng ngả về phía mặt nước - Qua thí nghiệm trên có kết luận: tia khúc xạ luôn nằm trong mặt phẳng tới. - Trong trường hợp ánh sáng đi từ nước ra ngoài không khí thì khi thay đổi góc tới góc khúc xạ thay đổi, nhưng góc khúc xạ luôn lớn hơn góc tới
8.	S#4- Thí nghiệm 4- Quan hệ giữa góc khúc xạ và chiết suất của hai môi trường	Bật loa	- Thí nghiệm 4-Quan hệ giữa góc khúc xạ và chiết suất của hai môi trường
		Nguồn sáng	- Đặt vị trí nguồn sáng
		Bình thuỷ tinh	- Đặt bình thuỷ tinh trong suốt
		Tấm gỗ	- Đặt tấm gỗ vuông góc với mặt nước
		Thước đo độ	- Đặt thước đo độ trên mặt phẳng phân cách
		Đổ nước vào bình	- Đổ dần dần nước vào bình thuỷ tinh
		Bật nguồn sáng	- Chiếu tia sáng từ không khí vào nước

		Dùng chuột điều khiển tấm gỗ	- Điều chỉnh tấm gỗ để tia tới và tia khúc xạ nằm là là trên mặt tấm gỗ
9.		Di chuyển thước đo độ	- Lấy thước đo độ đo góc tới và góc khúc xạ
		Chọn chiết suất khác: nước, dầu, thủy tinh, chất dẻo	- Thay đổi môi trường trong suốt thứ hai, góc khúc xạ thay đổi
	Mô hình hóa thí nghiệm 4	Bật loa	- Hình ảnh mô tả quan hệ giữa góc khúc xạ và chiết suất của hai môi trường - Lần lượt thay đổi môi trường trong suốt thứ 2, góc khúc xạ thay đổi tương ứng. - Qua thí nghiệm trên có kết luận: khi ánh sáng truyền từ không khí vào các môi trường trong suốt khác nhau thì góc khúc xạ có độ lớn khác nhau

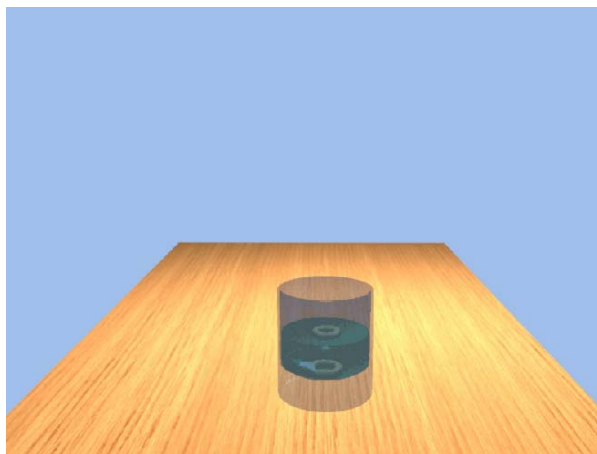
d. Video

Chương trình sử dụng 5 đoạn phim ngắn, các đoạn phim này đều được xây dựng trên 3D Studio Max và Adobe Premiere.

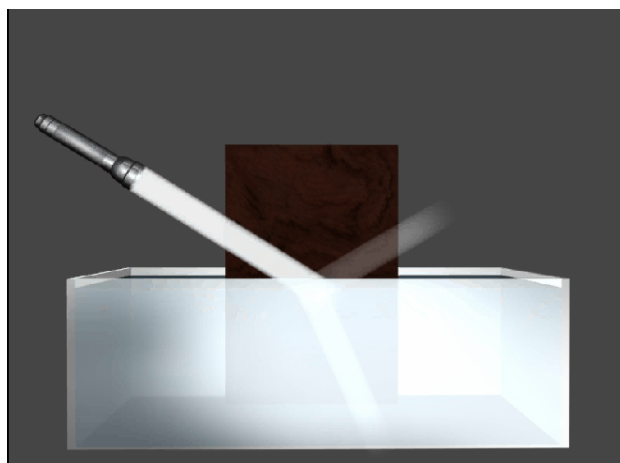
Video1- Mô tả hiện tượng chiếc đũa cắm vào cốc thủy tinh- chiếc đũa thẳng. Khi đổ nước vào cốc, chiếc đũa hình như bị gãy.



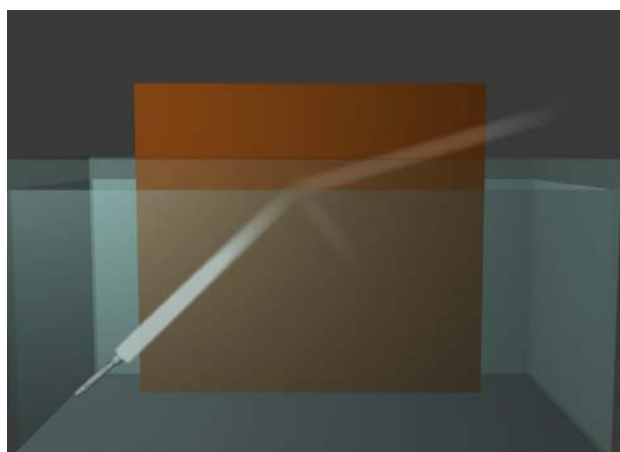
Video 2- mô tả hiện tượng nhìn cái vòng bạc dưới đáy một cốc nước lớn, hình ảnh chiếc vòng bạc hình như ở gần mặt nước hơn.



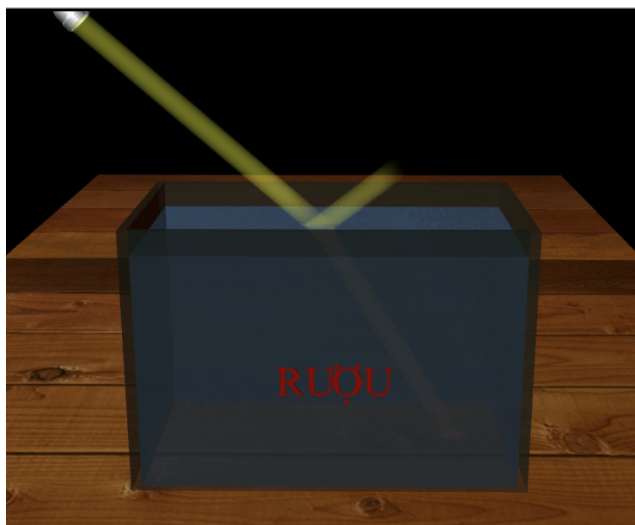
Video 3- mô hình hóa thí nghiệm biểu diễn đường đi của tia sáng từ không khí vào nước. Chiều chùm sáng từ không khí vào mặt nước, đến mặt nước chùm sáng đổi hướng đi vào nước.



Video 4- mô hình hóa thí nghiệm biểu diễn đường đi của tia sáng từ nước ra ngoài không khí.



Video 5- mô hình hóa thí nghiệm biểu diễn quan hệ giữa góc khúc xạ và chiết suất của hai môi trường



1.1.1 Lập trình

Qua việc đánh giá hiện trạng và lựa chọn kỹ thuật chúng tôi quyết định sử dụng phần mềm Director và ngôn ngữ lập trình Lingo để xây dựng sản phẩm theo đúng như kịch bản đã đề ra.

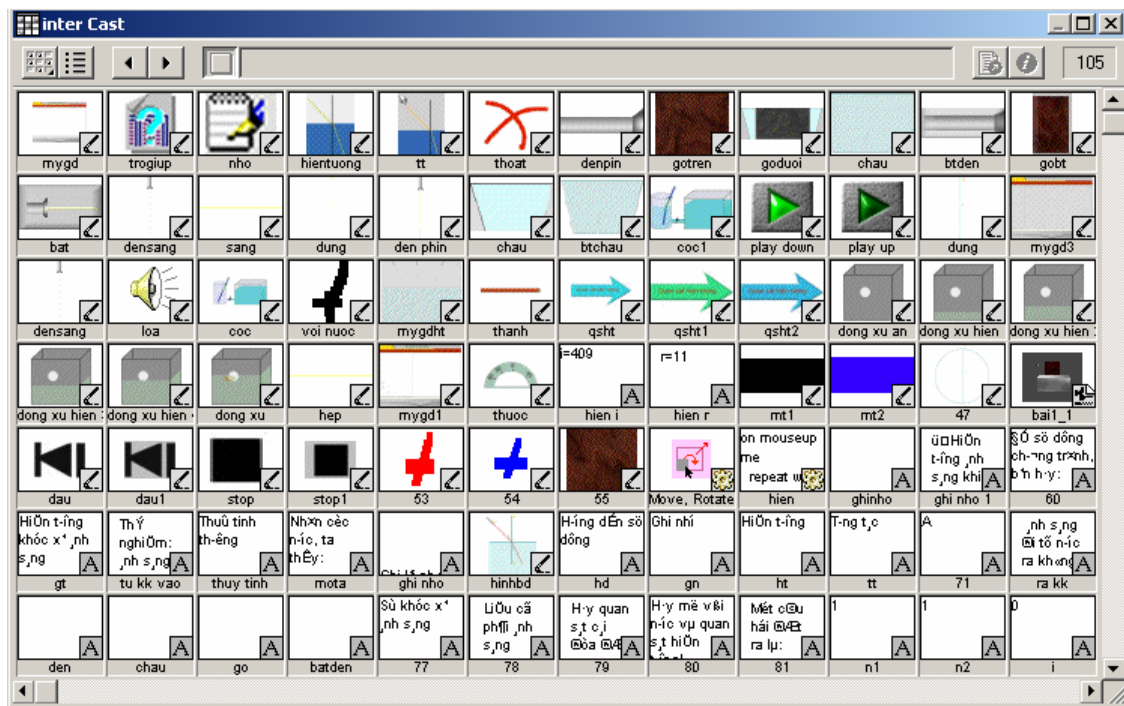
Phần mềm Director của hãng Macromedia là một công cụ sáng tác có tiềm lực công nghiệp mạnh, được thiết kế để phát triển các ứng dụng đa phương tiện chuyên nghiệp.

Director có thể được sử dụng để tạo ra các ứng dụng đa phương tiện có nhiều mục đích sử dụng khác nhau như: các tạp chí trực tuyến, các catalog sản phẩm, các công cụ hỗ trợ điện tử, các ứng dụng mô phỏng, chương trình quảng cáo, trò chơi giải trí và đặc biệt là các chương trình dạy học trên máy tính (computer based training)... chạy trong môi trường Windows, Web hay các máy Macintosh.

Chúng tôi đã sử dụng các chức năng và tài nguyên của Director như sau:

Cửa sổ Cast

Chúng tôi lưu tất cả dữ liệu đã xây dựng để tạo file .DIR, cửa sổ Cast hiển thị các thành phần mà ta nhập từ bên ngoài hoặc tạo ra- gọi là cast members. Chúng có thể là văn bản, hình ảnh, âm thanh,... và các file movie khác. Đặc biệt cast members ở đây có thể là những đối tượng mà trong Score và State không có, chẳng hạn như: kịch bản, bảng màu, phong chữ. Không chỉ nhập dữ liệu từ bên ngoài mà Director còn cho phép ta tạo các cast members như: ảnh vector, văn bản, behavior inspector, ... Cửa sổ cast cho phép chúng ta hiển thị các cast members bằng 2 cách-tùy theo sở thích: danh sách các cast member hoặc sự mô tả ngắn gọn các cast members.

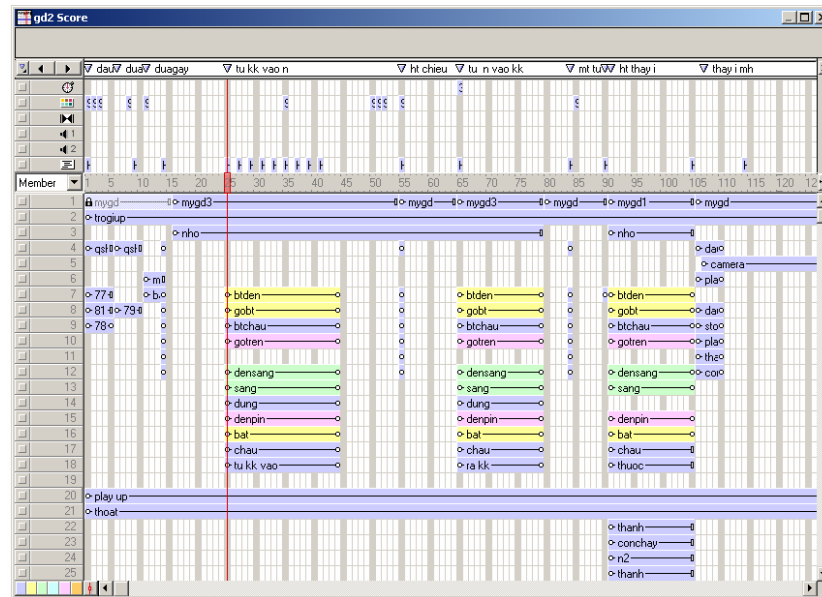


Trong đó danh sách các biểu tượng của cast members trong cửa sổ Cast như sau:

Biểu tượng	Kiểu Cast member	Biểu tượng	Kiểu Cast member
	Animated GIF		Behavior
	Bitmap		Button
	Check box		Custom Cursor
	Digital video		Field
	Film loop		Font
	Flash movie		Linked bitmap
	OLE		Palette
	PICT		QuickTime video
	Radio button		Script
	Shape		Shockwave Audio
	Transition		Vector shape
	Xtra		

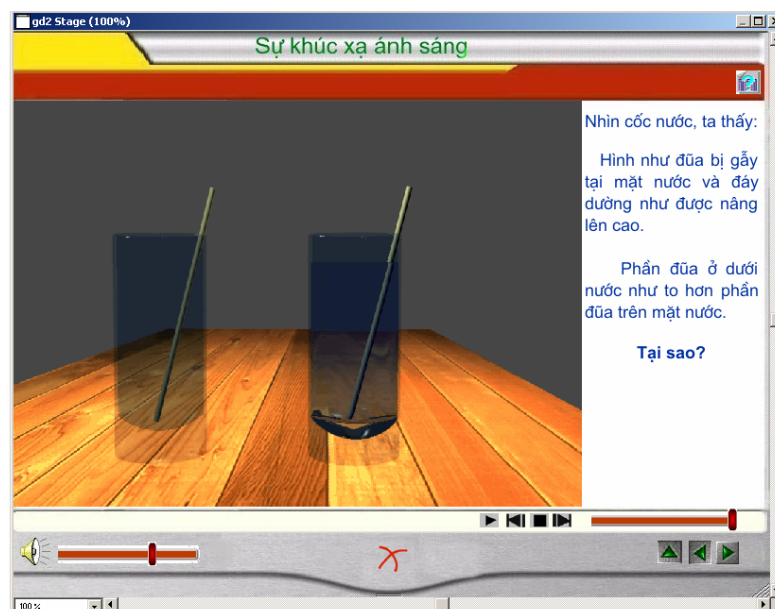
Cửa sổ Score

Score được xem như một bàn trộn phim, từ cửa sổ này chúng ta thiết lập nội dung và các điều khiển trong file.DIR thông qua các kênh (channels). Trong Score còn có các kênh đặc biệt điều khiển tốc độ, âm thanh và bảng màu. Thêm vào đó chúng ta cần quan tâm đến frame và thanh chạy (playback head). Chúng ta sử dụng Score để gán kịch bản – chỉ thị lingot để tạo ra những sự kiện trong file.DIR.



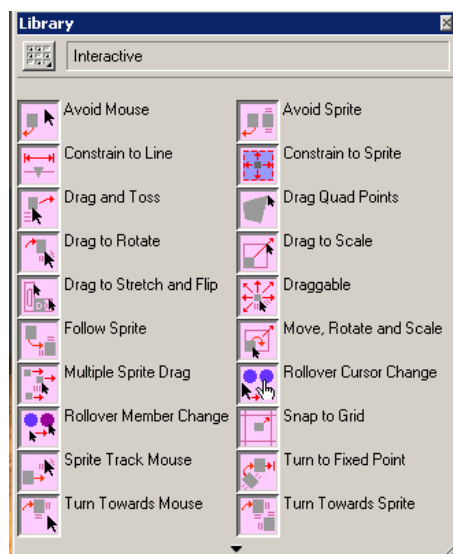
Cửa sổ State

State là nơi trình chiếu tất cả những sự kiện và hình ảnh mà có ở trong Score. Hay nói cách khác State là nơi hiển thị chương trình mà chúng ta muốn gửi đến người sử dụng. Ví dụ, ta kéo một sound member từ cửa sổ Cast vào trong sound channel của Score và một text member vào trong channel thì khi chạy file.DIR ta sẽ nhìn thấy văn bản hiện trong State và nghe âm thanh cùng một thời điểm.



Library palette

Director cung cấp cho chúng ta một số behavior đơn giản được đặt trong library palette. Các behavior này mang tính chất sẵn dùng, có nghĩa là chúng ta chỉ việc kéo và thả chúng vào *sprites/channel script* trong Score (hoặc *frame* trong State).



Mỗi behavior có một tính năng khác nhau. Ví dụ, khi ta kéo biểu tượng của behavior có tên là *Hold on Current Frame* vào một vị trí nào đấy trong channel script thì kết quả là playback head sẽ dừng lại khi chạy đến vị trí đấy. Một ví dụ khác, khi ta kéo biểu tượng của *play sound* vào trong một đối tượng nào đấy trong State thì sẽ hiện ra bảng thông báo cho ta chọn chế độ kích hoạt và lựa chọn một trong số sound member có trong cửa sổ Cast. Khi chạy file.DIR, ta kích hoạt vào đối tượng thì sẽ nghe âm thanh của sound member mà chúng ta chọn.

Lingo scripts- mã kịch bản của Director

Trừ các behavior sẵn có đặt trong library palette, lingo scripts chính là công cụ cho phép chúng ta tạo các behavior mới để thêm một số tương tác đến file. DIR. Ta có thể sử dụng lingo để điều khiển một đối tượng trong file.DIR trả lời các tình huống và sự kiện đặc biệt. Ví dụ, lingo có thể tạo các đoạn văn bản có khả năng liên kết (hyperlink text), hay thậm chí làm các phép toán cộng, trừ, nhân, chia, ...

Lingo có 7 loại: lệnh (command), đặc tính (property), hàm (function), từ khoá (keyword), sự kiện (event), constant (hằng số), và toán tử (operator). Trong việc sử dụng

danh sách để quản lý dữ liệu, lingo có thể tạo mới, khôi phục lại, thêm vào, sắp xếp, phân loại hoặc thay thế nội dung của một danh sách.

Director sử dụng 4 kiểu mã kịch bản (script), chúng gồm mã kịch bản trong các behavior có sẵn hoặc do người dùng tạo ra, movie script, parent script, mã kịch bản gắn với cast member. Chúng xuất hiện như là cast member trong cửa sổ Cast:

Đơn giản và dễ tìm nhất là các behavior có sẵn trong library palette. Chúng gắn với sprites hoặc frames trong Score và được xem như là behavior sprites hoặc behavior frames. Chúng ta có thể gắn cùng một behavior đến nhiều nơi trong Score. Nếu behavior có tham biến thì hộp thoại sẽ xuất hiện để ta lựa chọn tham biến.

Nếu như một đối tượng có chứa nhiều sự kiện, thay vì phải gắn cho đối tượng đó nhiều behavior sẵn có trong library palette ta có thể tạo một behavior có chứa đồng thời nhiều sự kiện bằng cách mở behavior inspector để tạo behavior mới cùng với các sự kiện và hoạt động của các sự kiện.

Kiểu thứ 2 là movie scripts, tại đây người dùng có thể tạo sự kiện (event) hoặc hoạt động (action) mới hoặc viết các hàm tác động đến toàn bộ file.dir. Các thẻ trong movie scripts có thể được gọi từ một mã kịch bản khác.

Ví dụ: khi chúng ta tạo hoạt động cho thao tác ấn chuột. Giả sử khi kích chuột vào một đối tượng (cast member) “A” trong cửa sổ State, chúng ta muốn hoạt động xảy ra là đối tượng text member “B” sẽ xuất hiện dòng chữ “Hello!”. Mã kịch bản sẽ là:

```
on text
    if member("b").text <> "" then
        member("b").text = ""
    end if
    member("b").text = "Hello !"
end
```

Parent scripts là một mã kịch bản đặc biệt mà nội dung của lingo nhằm để tạo ra các đối tượng con. Parent scripts mang lại sự thuận lợi cho việc lập trình hướng đối tượng trong Director. Nó giúp cho việc viết mã kịch bản trở nên ngắn hơn và suy luận logic đơn giản hơn. lingo có thể tạo ra nhiều bản copy của một mã nguồn (parent script), mỗi bản này được gọi là các đối tượng con. Director không hạn chế số đối tượng con có thể được tạo ra từ cùng một mã nguồn. Có nghĩa là Director có thể tạo ra nhiều trường hợp của một

behavior cho các sprite khác nhau. Vậy ta có thể nói rằng một mã nguồn như là một template (mẫu) và mã kế thừa như một sự bổ sung cho template nguồn (parent template).

Ví dụ1: Để đặt một biến thuộc tính cho đối tượng ta sử dụng từ khoá *property*

Cú pháp: *property* [property1], [property2], ...

Ví dụ2: kiểm tra thuộc tính của đối tượng :

Cú pháp: <tên_đối tượng>.<tên_thuộc_tính>

Chẳng hạn, khi ta gán cho biến x thuộc tính carSpeed của đối tượng con trong biến car1. Ta có:

```
x = car1.carSpeed
```

Đoạn mã sau đây kiểm tra xem nếu đối tượng car1 là đối tượng con của Car thì sẽ có tiếng beep phát ra:

```
if car1.script = script("Car") then
    beep
end if
```

Ví dụ3: Để tạo một đối tượng con ta sử dụng cú pháp:

Để tạo đối tượng con, bạn sử dụng hàm new và gán cho đối tượng con một biến tên hoặc đặt nó ở vị trí nào đó trong một danh sách để sau này bạn có thể nhận ra và làm việc với nó.

Khi ta muốn tạo đối tượng con và gán nó cho một biến tên, sử dụng cú pháp:

```
set variableName = new(script "scriptName", argument1,
    argument2, argument3...)
```

ScriptName là tên của đối tượng cha và argument1, argument2, ... là các đối số mà ta thông qua chúng để đặt vào các chỉ thị mới cho đối tượng con. Hàm new() tạo một đối tượng con của scriptName.

Director còn cung cấp cho chúng ta hàm rawNew() để tạo đối tượng con mà không cần khởi tạo cho nó các biến thuộc tính. Chẳng hạn, khi ta muốn tạo một đối tượng con của đối tượng cha Car và gán nó cho biến car1:

```
car1 = script("Car").rawNew()
```

Để khởi tạo thuộc tính của một trong những đối tượng con, gọi chúng trong chỉ thị new:

```
car1.new
```

Mã kịch bản gắn với cast member: Đây là việc gắn trực tiếp mã kịch bản với cast member, không phụ thuộc Score. Khi nào cast member được gắn vào một sprite thì mã kịch bản của cast member được thi hành.

Không như behavior, movie scripts, parent scripts, cast member scripts không xuất hiện trong cửa sổ Cast. Khi nào nhấn chuột vào hộp thoại cast member properties của mỗi đối tượng cast member hoặc mở cast member script từ cửa sổ script.

Dựa theo các cảnh trong kịch bản chúng ta có thể chia thành 4 module chính của chương trình: hai hiện tượng quan sát sự khúc xạ ánh sáng, Thí nghiệm và mô hình hóa hiện tượng khúc xạ ánh sáng, khảo sát hiện tượng khúc xạ ánh sáng và mô hình hóa kết quả thí nghiệm, quan hệ giữa góc khúc xạ và chiết xuất của hai môi trường.

Dựa theo kịch bản đã được viết để chúng ta xây dựng module chương trình. Công việc này xem như có vẻ dễ dàng hơn khi mà trong tay chúng ta có kịch bản. Điều khó khăn nhất của chúng ta bây giờ là làm sao gắn cho các đối tượng các chỉ thị như trong kịch bản đã đề ra. Tạo các chỉ thị tương đương với việc gắn cho các đối tượng *behavior* tương ứng hoặc viết mã kịch bản trong *movie script*, kịch bản gắn với *cast member*. Trừ một số behavior sẵn có trong library palette, chúng ta sẽ phải tự tạo các behavior bằng cách viết mã kịch bản.

Chúng tôi đã xây dựng 35 đoạn mã lệnh khác nhau để có thể cho phép người dùng tương tác với các đối tượng trong thí nghiệm. Để một lệnh tác động đến toàn bộ chương trình hoặc một đối tượng nào đó trong chương trình mà ta không cần sử dụng thao tác gắn *behavior* vào đối tượng thì ta cần phải viết lệnh ở trong *movie script*.

Chẳng hạn, để khi chuyển sang cảnh khác, ta muốn âm thanh của cảnh cũ sẽ tắt và đồng thời có tín hiệu của tệp âm thanh đặt trong cảnh mới, ta có đoạn mã:

```
on enterframe
  Set the soundEnabled=True

End enterframe
on exitFrame
  Set the soundEnabled=False
```

```

        Set the soundEnabled=True
    End exitFrame
on startmovie
    Set the soundEnabled=False
    Set the soundEnabled=True
End startmovie

```

Để chương trình đọc và hiện dòng thông báo khi ta bấm chuột vào dòng chữ liên kết:

```

on cau1
    - -câu 1 là tên được đặt khi ta định nghĩa một nhóm từ trong văn -
    - bản là nhóm từ liên kết.
    sprite(17).member = member("cuoi")
    sound playfile 1, "../soundt/good.wav"
    loop=1
end

```

Còn khi nhập xâu kí tự vào ô trống, sau thao tác bấm chuột vào một đối tượng, chương trình sẽ đưa ra thông báo đúng hoặc sai thì ta viết kịch bản gắn với đối tượng đó:

```

on mousedown
    tu_dua_vao=member("dientu").text
    if tu_dua_vao = "mouse" then alert "dung"
    else alert "sai"
    end if
end

```

Trong quá trình nghe bài hội thoại, khi bấm chuột vào một đối tượng thì bài đọc ngừng lại và bấm tiếp một lần nữa vào đối tượng đấy thì bài đọc sẽ tiếp tục:

```

on mousedown
    soundStatus = sound (1).status
    if soundStatus = 4 then
        sound (1).play ()
    else
        sound (1).pause()
    end if
end

```

Hết cảnh video này chương trình sẽ tự động chuyển vào trang tiếp theo. Trong trang hiện tượng hoặc thí nghiệm, người học dựa vào các mối liên kết - đối tượng liên kết là tiêu đề của các bài học mà có thể chọn các đối tượng và tương tác với các đối tượng để quan sát hiện tượng một cách tùy ý. Khi xem các đoạn video mô hình hóa thí nghiệm

người học có thể tắt hoặc tạm dừng tùy ý. Trong màn hình nhận xét, mục đích là để người học xem hình ảnh minh họa và đọc hiểu nội dung của thí nghiệm là gì do vậy người học có thể bấm vào nút Back để trở về ngay hiện tượng hay thí nghiệm mà mình vừa quan sát để hiểu rõ hơn nội dung này... Để hiện hình ảnh khi kích hoạt vào đối tượng ta có thể làm bằng 2 cách, hoặc là viết đường dẫn đến file liên kết vào trong khung *hyperlink data*, hoặc là chỉ đưa vào một nhóm đối tượng rồi sau đó định nghĩa nhóm đối tượng bằng mã kịch bản viết trong *movie script*.

Ví dụ:

On từ_tự_đặt

Sprite(14).member = member("computer")

End

Cụ thể trong chương trình chúng tôi đã xây dựng các Script và Behaviour sau:

1. my Drag to Rotate: kéo rê chuột điều chỉnh góc quay của đối tượng

```
global newAngle,n1,n2,i,r
-- PROPERTIES --
```

```
property spriteNum
property mySprite
property myMember
```

```
-- internal properties
property myMouseDown
property myPrevAngle
property myClickAngle
```

```
-- EVENT HANDLERS --
```

```
on beginSprite me
  Initialize me
end beginSprite
```

```
on mouseDown me
```

```
  StartDrag me
end mouseDown
```

```
on mouseUpOutside me
```

```

    myMouseDown = FALSE
end mouseUpOutside

on prepareFrame me
    if myMouseDown then
        if the mouseDown then
            Turn me
        else
            StopDrag me
        end if
    end if
end prepareFrame

-- CUSTOM HANDLERS --

on Initialize me -- sent by beginSprite
    mySprite = sprite(me.spriteNum)
    myMember = mySprite.member
    memberType = myMember.type

end Initialize

on Forwarded_mouseDown me
    -- The mouseclick was intercepted by a different behavior and
    forwarded
    if not myMouseDown then
        mouseDown me
        return myMouseDown
    end if
end mouseUp

```

```

on StartDrag me -- sent by mouseDown
  if sprite (spriteNum).ink = 8 then
    -- Matte ink
    if the mouseMember <> sprite (spriteNum).member then
      exit
    end if
  end if
  --if (newangle>=180) and(newangle<360) then
  myMouseDown = TRUE
  sendAllSprites #Active_Sprite, myMouseDown
  slope = the clickLoc - mySprite.loc
  myClickAngle = GetAngle (me, slope)
  myPrevAngle = mySprite.rotation
end StartDrag

on StopDrag me
  myMouseDown = FALSE
  sendAllSprites #Active_Sprite, myMouseDown
end StopDrag

on Turn me -- sent by prepareFrame
  slope = the mouseLoc - mySprite.loc
  newAngle = myPrevAngle + getAngle (me, slope) - myClickAngle
  if (newAngle> 0) and (newangle<90) then newangle=360
  if (newangle<270) and(newangle>90) then newangle=270
  if newangle>360 then newangle=360
  mySprite.rotation = newAngle

  newset1 me, newAngle
end Turn

on newset1 me , goc
  i=360-newangle
  n1=1
  if n2<>1 then
    if i<>0 then
      sinr=n1*sin(i*PI/180)/n2
      cosr=sqrt(1-sinr*sinr).float
      r=(atan(sinr/cosr)*180/PI ).integer
      sprite(13).width=sprite(17).height/cosr
    else
      r=0
    end if
  else
    r=i
  end if
  sprite(13).rotation=90-r
  --sprite(11).loc=point(
  sprite(12).lochsprite(12).height*abs(sin(i)), sprite(12).locv-
  sprite(12).height*abs(cos(i)))
end

```

```

on GetAngle me, slope -- sent by Turn
    deltaH = slope[1]
    deltaV = slope[2]
    if deltaH then
        slope = float (deltaV) / deltaH
        angle = atan (slope)
        if deltaH < 0 then
            angle = angle + pi
        end if
    else if deltaV > 0 then
        angle = pi / 2
    else if deltaV < 0 then
        angle = (3 * pi) / 2
    else
        angle = 0
    end if
    -- Convert to degrees for .rotation
    angle=angle*180/pi
    return angle
end GetAngle

-- ERROR CHECKING --

on isOKToAttach (me, aSpriteType, aSpriteNum)
    case aSpriteType of
        #graphic:
            return getPos([#bitmap, #flash, #picture, #text, #vectorShape],
sprite(aSpriteNum).member.type) <> 0
        #script:
            return FALSE
    end case
end isOKToAttach

on getPropertyDescriptionList me

    if not the currentSpriteNum then exit

end getPropertyDescriptionList

```

1. an bat den: điều khiển đối tượng nguồn sáng tắt mở

```

on mouseup me
    go to frame 23
end

```

2. an chau: điều khiển hiển thị đối tượng chậu

```

on mouseup me

```

```
go to frame 21
end
```

3. *ango: điều khiển hiển thị đối tượng tám gố*

```
on mouseup me
go to frame 19
end
```

4. *batden: bật tắt đèn*

```
global i, r, n1, n2

on mouseup me
repeat with j=11 to 14 do
sprite(j).visible=true
end repeat
i=30
-- sprite(19).visible=true
sprite(15).visible= false
if sprite(17).visible=false then
r=i
n2=1
else
sprite(18).visible=true
r=22
end if
sprite(13).rotation=90-r
end
```

5. *begin: khởi tạo chương trình*

```
property pchannel,
on mousedown me
sprite(pchannel).movietime=1
sprite(pchannel).movieRate=1
end

on getPropertyDescriptionList me
set mylist=[]

addProp mylist, # pChannel, [#default: 2, #format: #integer,
#range: [#min: 1, #max:70], #comment: "member channel"]
return mylist
end
```

6. *chon cơ sở dữ liệu: chọn màn hình hiển thị*

```
global n4, default
property mysp

on mouseup me
mysp=sprite(me.spritenum)
```

```

n4=(mysp.locH-sprite(22).locH)*3/sprite(22).width+1
-- sprite(23).locH = (n4-1) * Sprite(22).width/3
+Sprite(22).locH
sprite(23).locH=mysp.locH
default=1
end

```

7. *closewin: đóng cửa sổ*

```

on mouseup me
  halt
end

```

8. *dieukhien sound: điều khiển âm thanh*

```

global gTargetChannel, gTargetChannel1
-- mn1, mn2
property pSprite
property pConstraintSprite

on beginSprite me
  pSprite = sprite(me.spriteNum)
  pSprite.moveableSprite = TRUE
  pConstraintSprite = sprite(me.spriteNum - 1)
  pSprite.constraint = pConstraintSprite
  pSprite.locH= pConstraintSprite.left + 100
  set the volume of sound gTargetChannel to 150
  set the volume of sound gTargetChannel1 to 50
end

on exitFrame me
  set the volume of sound gTargetChannel to (pSprite.locH -
pConstraintSprite.locH)*255.0/pConstraintSprite.width+50
  set the volume of sound gTargetChannel to the volume of sound
gTargetChannel-50
  pSprite.locV= pConstraintSprite.locV
end

on ChangeTarget me
  pSprite.locH = the volume of sound gTargetChannel *
pConstraintSprite.width/255.0 + pConstraintSprite.locH
  pSprite.locV= pConstraintSprite.locV
end

```

9. *Display Text: hiển thị dòng văn bản trên màn hình*

```

-- DESCRIPTION --

on getBehaviorDescription me
  return \
  "DISPLAY TEXT" & RETURN & RETURN & \

```

```

    "This behavior allows you to display a given string in a field
    or text member. " & \
    "Use it with the Tooltip and Hypertext - Display Status
    behaviors which need a field or text member in which to display
    their information. " & \
    "Or create your own custom Lingo to display runtime information,
    such as the position of the mouse." & RETURN & RETURN & \
    "This behavior waits for Lingo commands to tell it what to do. "
    & \
    "It is not active by itself." & RETURN & RETURN & \
    "You can choose between two display types: tooltip and status
    bar." & RETURN & RETURN & \
    "The TOOLTIP type of display will make the field or text member
    resize itself to fit the text, and disappear when it is empty. "
    & \
    "You can set the tooltip type display to appear at any position
    on the stage, such as under the cursor. " & \
    "If no position is sent to the sprite, it will appear at the top
    left corner of the Stage. " & \
    "See the Tooltip behavior for more details." & RETURN & RETURN &
    \
    "If you wish to display several lines of text, you must use
    RETURN characters to define the line breaks. " & \
    "An empty tooltip sprite will move off-stage to hide. " & \
    "It is recommended that you place it off-stage before it is
    used, in case it causes a brief flash on the screen." & RETURN &
    RETURN & \
    "The STATUS BAR type of display will appear on Stage at all
    times. " & \
    "It will not resize or change position. " & \
    "Any positional information sent to this sprite will be ignored
    if it is set to act as a status bar. " & \
    "If the text is too long to appear in the member of the current
    sprite, a scrollbar will appear. " & \
    "You do not need to divide the text with RETURN characters. " &
    \
    "If you think that a scrollbar may be necessary, make sure that
    the field or text member is sufficiently tall for the scroll
    arrows to operate correctly." & RETURN & RETURN & \
    "Set the font size and other characteristics of the field or
    text member to customize the appearance of the message." & RETURN
    & RETURN & \
    "Be sure to give the field or text member a name. " & \
    "It may be emptied by this behavior. " & \
    "Director automatically erases nameless empty members." & RETURN
    & RETURN & \
    "PERMITTED MEMBER TYPES:" & RETURN & \
    "field and text" & RETURN & RETURN & \
    "PARAMETERS:" & RETURN & \
    "* Display type:" & RETURN & \
    " - Tooltip (appears near the cursor on rollover)" & RETURN & \
    " - Status bar (appears in a fixed position at all times)" &
    RETURN & RETURN & \
    "PUBLIC METHODS:" & RETURN & \

```

```

    "* Set the text to display (and the position of the sprite)" &
RETURN & RETURN & \
    "ASSOCIATED BEHAVIORS:" & RETURN & \
    "* Tooltip" & RETURN & \
    "* Source Status" & RETURN & \
    "* Hypertext - Display Status"
end getBehaviorDescription

on getBehaviorTooltip me
    return \
        "Use with field or text members." & RETURN & RETURN & \
        "Waits for a message from another behavior or custom handler to
display a character string. " & \
        "This behavior is intended to be used with the Tooltip and
Hypertext - Display Status behaviors to create a status bar or a
tooltip under the cursor."
end getBehaviorTooltip

-- NOTES FOR DEVELOPERS --

-- COMMUNICATING WITH THE SPRITE
-- To set the text of the current sprite's member, use a line
similar to one of
-- the following:
--
-- sendAllSprites (#DisplayText_SetText, theStringToDisplay)
--
-- sendSprite (<spriteNumber>, #DisplayText_SetText,
theStringToDisplay)
--
-- call (#DisplayText_SetText, displayBehavior,
theStringToDisplay)
--
-- It is fastest to call the behavior directly. Use code similar
to the
-- following lines in any script that needs to communicate with
this behavior
-- often:
--
-- property displayBehavior
--
-- ...
-- displayBehavior = []
-- sendAllSprites (#DisplayText_Enroll, displayBehavior)
-- ...
--
-- If your _Enroll call fails for some reason, displayBehavior
will be an
-- empty list. Calling an empty list will not produce an error
(calling an
-- invalid behavior reference will). It would, however, be wise
to provide

```

```

-- yourself with an authortime alert so that you can correct such
a problem
-- if necessary:
--
-- if displayBehavior.count() = 0 then
-- if the runMode = "Author" then
-- alert "No 'Display Text' behavior found in frame "the frame
-- end if
-- end if
--
-- Writing this sort of "defensive" code should ensure that any
bugs that make
-- it through to the finished product are relatively harmless.

-- ADJUSTING THE WIDTH OF THE SPRITE
-- The BestRect function is called if you choose the Tooltip
display type. It
-- modifies the width of the sprite to suit the longest
line/paragraph of text \
-- (field lines and text paragraphs are delimited by the RETURN
character).
--
-- This function uses the charPosToLoc() function to determine
the length of
-- each line. CharPosToLoc returns the position of the bottom
left corner of
-- the tested character: if the character tested is the final
RETURN of a line,
-- then the value returned is equiavalent to the bottom right of
the last
-- visible character in the line.
--
-- I initially set the width of the member to an extravagantly
large value, to
-- ensure that no line-wrapping should occur.

-- HISTORY --

-- 1 october 1998: Written for the D7 Behaviors Palette by James
Newton
-- 28 October 1998: Descriptions improved. getPDL simplified,
GetTopLeft() added
-- 7 January 2000: Added isOKToAttach and substituteStrings event
handlers
-- and removed redundant error checking code - Karl Miller
-- 13 January 2001: Set state of member to not editable on
initialize. Reset on endSprite
-- Set the locZ to maxInt - 1 to place the tooltip on top. Reset
on endSprite
-- -- Kraig Mentor

```

```

-- PROPERTIES --

property spriteNum
-- error checking
property getPDLError
-- author-defined parameters
property myDisplayType
-- internal properties
property mySprite
property myMember
property myWidthAdjust
property myHeightAdjust
property myOffStageLoc
property myOriginalEditableState
property myOriginalLocZ

-- EVENT HANDLERS --

on beginSprite me
    myDisplayType = resolve(myDisplayType)
    Initialize me
end beginSprite

on endSprite me
    mySprite.visible = TRUE
    myMember.editable = myOriginalEditableState
    mySprite.locZ = myOriginalLocZ
end endSprite

-- CUSTOM HANDLERS --

on resolve(prop)
    case prop of
    myDisplayType:
        choicesList = ["status bar (fixed size and position)", \
            "tooltip (dynamic size and position)"]
        lookup = [#statusBar, #tooltip]
    end case
    return lookup[findPos(choicesList, prop)]
end resolve

on Initialize me -- sent by beginSprite

    mySprite = sprite(me.spriteNum)
    myMember = mySprite.member
    myOriginalEditableState = myMember.editable
    myOriginalLocZ = mySprite.locZ
    myMember.editable = FALSE

    if myMember.type = #field then

```

```

myWidthAdjust = (myMember.margin + myMember.border) * 2
myHeightAdjust = myMember.margin + myMember.border * 2
-- no gremlins here:)
else
myWidthAdjust = 0
myHeightAdjust = 0
end if
myMember.text = EMPTY

if myDisplayType = #tooltip then
myMember.boxType = #fixed
myOffStageLoc = point (999, 999)
mySprite.loc = myOffStageLoc
end if
end Initialize

on BestRect me, theString -- sent by DisplayText_SetText
-- Sets the rect of myMember to fit snugly round theString it
displays

myMember.rect = rect (0, 0, 8000, 0)
myMember.text = theString -- Needed to update myMember.rect
bestRect = myMember.rect
theLine = the number of lines of theString
theWidth = 0
checkedChars = 0
-- Determine the length of the longest line
repeat while theLine
endOfLine = offset (RETURN, theString)
if not endOfLine then
-- Only one line remaining
endOfLine = (the number of chars of theString) + 1
myMember.text = myMember.text & RETURN
end if
checkedChars = checkedChars + endOfLine
endPoint = charPosToLoc (myMember, checkedChars)
lineWidth = endPoint[1]
if lineWidth > theWidth then
theWidth = lineWidth
end if
delete char 1 to endOfLine of theString
theLine = theLine - 1
end repeat
-- Determine the height of the text
lastChar = myMember.char.count
lastCharLoc = charPosToLoc (myMember, lastChar)
theHeight = lastCharLoc[2]
--
bestRect[3] = theWidth + 1
bestRect[4] = theHeight + 1
return bestRect

```

```
end BestRect
```

```
on GetTopLeft me, theLoc, theAlignment, memberRect
  case theAlignment of
    #bottomCenter: return theLoc - [memberRect.width / 2,
memberRect.height]
    #bottomRight: return theLoc - [memberRect.width,
memberRect.height]
    #bottomLeft: return theLoc - [0, memberRect.height]
    #center: return theLoc - [memberRect.width / 2,
memberRect.height / 2]
    #topCenter: return theLoc - [memberRect.width / 2, 0]
    #topRight: return theLoc - [memberRect.width, 0]
    otherwise -- treat as #topLeft
  return theLoc
end case
end GetTopLeft
```

```
-- INTER-SPRITE COMMUNICATION (response to #sendSprite,
#sendAllSprites) --
```

```
on DisplayText_Enroll me, enrollList
  -- sent by objects which need to call this specific behavior
  if ilk (enrollList) <> #list then return me
  if not enrollList.count() then
    enrollList.append(me)
  else
    -- the calling behavior has already found a candidate
  end if
  return enrollList
end DisplayText_Enroll
```

```
on DisplayText_SetText me, theString, theLoc, theAlignment
  -- called by other objects
  -- Sets the text of myMember to theString and shows mySprite
  -- as near to theLoc as possible (if myDisplayType is #tooltip)
  --
  -- theAlignment can take any of the following values:
  --
  #bottomCenter|#bottomRight|#bottomLeft|#center|#topCenter|#topRig
ht|#topLeft
  -- This determines which point of the current sprite is to
  appear at theLoc

  -- Error check
  if not stringP (theString) then
    ErrorAlert (me, #invalidString, theString)
    theString = string (theString)
  else
    case ilk (theLoc) of
```

```

#void, #point: -- nothing
otherwise
ErrorAlert (me, #invalidPoint, theLoc)
theLoc = point (0, 0)
end case
end if
-- End of error check

if theString = EMPTY and myDisplayType = #tooltip then
mySprite.loc = myOffStageLoc
mySprite.locZ = myOriginalLocZ
else

myMember.text = theString
if myDisplayType = #tooltip then
memberRect = BestRect (me, theString)
myMember.rect = memberRect
else
memberRect = myMember.rect
end if
memberRect = memberRect + [0, 0, myWidthAdjust, myHeightAdjust]
if myDisplayType = #tooltip then
mySprite.locZ = the maxInteger

if ilk (theLoc) <> #point then
theLoc = point (0, 0)
end if
theLoc = GetTopLeft (me, theLoc, theAlignment, memberRect)
-- Ensure sprite is fully visible on stage
stageWidth = (the activeWindow).rect.right - (the
activeWindow).rect.left
stageHeight= (the activeWindow).rect.bottom - (the
activeWindow).rect.top
maxH = stageWidth - memberRect.width
maxV = stageHeight - memberRect.height
theLoc[1] = max (0, min (theLoc[1], maxH))
theLoc[2] = max (0, min (theLoc[2], maxV))

theLoc = theLoc + myMember.regPoint
mySprite.loc = theLoc
else
lastChar = theString.char.count
textHeight = charPosToLoc (myMember, lastChar)[2]
if textHeight > mySprite.height then
myMember.boxType = #scroll
else
myMember.boxType = #fixed
end if
end if
end if
end DisplayText_SetText

```

```

on DisplayText_GetReference me
    return me
end DisplayText_GetReference

-- ERROR CHECKING --

on ErrorAlert me, theError, data
    -- Determine the behavior's name
    behaviorName = string (me)
    delete word 1 of behaviorName
    delete the last word of behaviorName
    delete the last word of behaviorName

    case theError of
    #invalidString:
        if the runMode = "Author" then
            message = substituteStrings(me, \
            "BEHAVIOR ERROR: Frame ^0, Sprite ^1" & RETURN & \
            "Behavior ^2" & RETURN & RETURN & \
            "The DisplayText_SetText handler could not treat the following
            as a string:" & RETURN & RETURN & \
            "^3", \
            ["^0": the frame, "^1": me.spriteNum, "^2": behaviorName, "^3":
            data])
            alert message
        end if
    #invalidPoint:
        if the runMode = "Author" then
            message = substituteStrings(me, \
            "BEHAVIOR ERROR: Frame ^0, Sprite ^1" & RETURN & \
            "Behavior ^2" & RETURN & RETURN & \
            "The DisplayText_SetText handler could not treat the following
            as a point:" & RETURN & RETURN & \
            "^3", \
            ["^0": the frame, "^1": me.spriteNum, "^2": behaviorName, "^3":
            data])
        end if
    end case
end ErrorAlert

on substituteStrings(me, parentString, childStringList) -----
-----
--
-- * Modifies parentString so that the strings which appear as
-- properties in childStringList are replaced by the values
-- associated with those properties.
--
-- <childStringList> has the format ["^1": "replacement string"]
-----
-----

```

```

i = childStringList.count()
repeat while i
tempString = ""
dummyString = childStringList.getPropAt(i)
replacement = childStringList[i]
lengthAdjust = dummyString.char.count - 1
repeat while TRUE
position = offset(dummyString, parentString)
if not position then
parentString = tempString&parentString
exit repeat
else
if position <> 1 then
tempString = tempString&parentString.char[1..position - 1]
end if
tempString = tempString&replacement
delete parentString.char[1..position + lengthAdjust]
end if
end repeat
i = i - 1
end repeat

return parentString
end substituteStrings

-- AUTHOR-DEFINED PARAMETERS --

on isOKToAttach (me, aSpriteType, aSpriteNum)
case aSpriteType of
#graphic:
return getPos([#field, #text], sprite(aSpriteNum).member.type)
<> 0
#script:
return FALSE
end case
end isOKToAttach

on getPropertyDescriptionList me

if not the currentSpriteNum then exit

return \
[ \
#myDisplayType: \
[ \
#comment: "Display Text sprite behaves as a", \
#format: #string, \
#default: "status bar (fixed size and position)", \
#range: ["status bar (fixed size and position)", \

```

```

    "tooltip (dynamic size and position)"] \
  ] \
]
end getPropertyDescriptionList

```

10. Drag to Rotate: kéo rê và quay đối tượng

```
-- DESCRIPTION --
```

```

on getBehaviorDescription me
  return \
    "DRAG TO ROTATE" & RETURN & RETURN & \
    "Use the mouse to drag a sprite in a circle around its
registration point." & RETURN & RETURN & \
    "PERMITTED MEMBER TYPES:" & RETURN & \
    "bitmap, Flash, picture, text, vector shape" & RETURN & RETURN & \
    "The scroll bar of scrolling Text members does not rotate." & \
    "If attached to a sprite with a scrolling Text member, this
behavior will convert the member to one with a fixed box size." &
RETURN & RETURN & \
    "PARAMETERS:" & RETURN & \
    "None"
end getBehaviorDescription

```

```

on getBehaviorTooltip me
  return \
    "Use with bitmaps, Flash, picture, text, and vector shape
members." & RETURN & RETURN & \
    "Allows a sprite to be dragged in a circle around its
registration point."
end getBehaviorTooltip

```

```
-- NOTES FOR DEVELOPERS --
```

```

-- Basically, this behavior compares the position of the mouse
when it was
-- clicked on the sprite and its current position. First the
slope between
-- the clickLoc and the loc of mySprite is converted to an angle.
Turn me
-- (called on each prepareFrame) then repeats the operation for
the current
-- mouseLoc and the current loc of mySprite. By subtracting one
angle from the
-- other, it knows how much the sprite should be turned from its
original
-- position.

```

```

-- HISTORY --

-- 14 September 1998: written for the D7 Behaviors Palette by
James Newton
-- 16 November 1998: ErrorAlert changed, made compatible with
Draw Connector


-- PROPERTIES --

property spriteNum
property mySprite
property myMember

-- internal properties
property myMouseDown
property myPrevAngle
property myClickAngle


-- EVENT HANDLERS --

on beginSprite me
  Initialize me
end beginSprite

on mouseDown me
  StartDrag me
end mouseDown

on prepareFrame me
  if myMouseDown then
    if the mouseDown then
      Turn me
    else
      StopDrag me
    end if
  end if
end prepareFrame


-- CUSTOM HANDLERS --

on Initialize me -- sent by beginSprite
  mySprite = sprite(me.spriteNum)
  myMember = mySprite.member
  memberType = myMember.type

```

```

end Initialize

on Forwarded_mouseDown me
    -- The mouseclick was intercepted by a different behavior and
    forwarded
    if not myMouseDown then
        mouseDown me
        return myMouseDown
    end if
end mouseUp

on StartDrag me -- sent by mouseDown
    if sprite (spriteNum).ink = 8 then
        -- Matte ink
        if the mouseMember <> sprite (spriteNum).member then
            exit
        end if
    end if

    myMouseDown = TRUE
    sendAllSprites #Active_Sprite, myMouseDown
    slope = the clickLoc - mySprite.loc
    myClickAngle = GetAngle (me, slope)
    myPrevAngle = mySprite.rotation
end StartDrag

on StopDrag me
    myMouseDown = FALSE
    sendAllSprites #Active_Sprite, myMouseDown
end StopDrag

on Turn me -- sent by prepareFrame
    slope = the mouseLoc - mySprite.loc
    newAngle = myPrevAngle + getAngle (me, slope) - myClickAngle
    mySprite.rotation = newAngle
end Turn

on GetAngle me, slope -- sent by Turn
    deltaH = slope[1]
    deltaV = slope[2]
    if deltaH then
        slope = float (deltaV) / deltaH
        angle = atan (slope)
        if deltaH < 0 then
            angle = angle + pi
        end if
    else if deltaV > 0 then
        angle = pi / 2
    else if deltaV < 0 then

```

```

    angle = (3 * pi) / 2
  else
    angle = 0
  end if
  -- Convert to degrees for .rotation
  angle = (angle * 180) / pi

  return angle
end GetAngle

-- ERROR CHECKING --

on isOKToAttach (me, aSpriteType, aSpriteNum)
  case aSpriteType of
    #graphic:
      return getPos([#bitmap, #flash, #picture, #text, #vectorShape],
        sprite(aSpriteNum).member.type) <> 0
    #script:
      return FALSE
  end case
end isOKToAttach

on getPropertyDescriptionList me

  if not the currentSpriteNum then exit

end getPropertyDescriptionList

```

11. Drag to Rotatecs: kéo rê và quay cùng lúc nhiều đối tượng

```

global newAngle ,n4,i2,r2, depen

```

```

-- PROPERTIES --

property spriteNum
property mySprite
property myMember

-- internal properties
property myMouseDown
property myPrevAngle
property myClickAngle

```

```

-- EVENT HANDLERS --

on beginSprite me
  Initialize me

```

```

end beginSprite

on mouseDown me
    depen = 1
    StartDrag me
end mouseDown

on mouseup me
    depen=0
end

on mouseUpOutside me
    depen=0
    myMouseDown = FALSE
end mouseUpOutside

on prepareFrame me
    if (depen=1) and myMouseDown then
        if the mouseDown then
            Turn me
        else
            StopDrag me
        end if
    end if
end prepareFrame

-- CUSTOM HANDLERS --

on Initialize me -- sent by beginSprite
    mySprite = sprite(me.spriteNum)
    myMember = mySprite.member
    memberType = myMember.type

end Initialize

on Forwarded_mouseDown me
    -- The mouseclick was intercepted by a different behavior and
    forwarded
    if not myMouseDown then
        mouseDown me
        return myMouseDown
    end if
end mouseUp

on StartDrag me -- sent by mouseDown
    if sprite (spriteNum).ink = 8 then
        -- Matte ink
        if the mouseMember <> sprite (spriteNum).member then
            exit
        end if
    end if
end StartDrag

```

```

end if
end if
--if (newangle>=180) and(newangle<360) then
myMouseDown = TRUE
sendAllSprites #Active_Sprite, myMouseDown
slope = the clickLoc - mySprite.loc
myClickAngle = GetAngle (me, slope)
myPrevAngle = mySprite.rotation
end StartDrag

on StopDrag me
myMouseDown = FALSE
sendAllSprites #Active_Sprite, myMouseDown
end StopDrag

on Turn me -- sent by prepareFrame
slope = the mouseLoc - mySprite.loc
newAngle = myPrevAngle + getAngle (me, slope) - myClickAngle
if (newAngle> 0) and (newangle<90) then newangle=360
if (newangle<270) and(newangle>90) then newangle=270
if newangle>360 then newangle=360
mySprite.rotation = newAngle
newset1 me, newAngle
end Turn

on newset1 me , goc

i2=360-newangle
Sprite(26).locH = i2+sprite(25).loch
if 1=n4 then
sprite(13).rotation=90-i2
r2=i2

else
if i2<>0 then
sinr= sin(i2*PI/180)/n4
if sinr<1 then
cosr=sqrt(1-sinr*sinr).float
r2=(atan(sinr/cosr)*180/PI ).integer

```

```

else
  r2=90

  end if
  else
  r2=0
  end if
  if r2<> 90 then
    sprite(13).rotation=90-r2
  else
    sprite(13).rotation=i2-90
  end if

  end if

  member("r").text=string(r2)

end

```

```

on GetAngle me, slope -- sent by Turn
  deltaH = slope[1]
  deltaV = slope[2]
  if deltaH then
    slope = float (deltaV) / deltaH
    angle = atan (slope)
    if deltaH < 0 then
      angle = angle + pi
    end if
  else if deltaV > 0 then
    angle = pi / 2
  else if deltaV < 0 then
    angle = (3 * pi) / 2
  else
    angle = 0
  end if
  -- Convert to degrees for .rotation
  angle=angle*180/pi
  return angle
end GetAngle

```

```

-- ERROR CHECKING --

```

```

on isOKToAttach (me, aSpriteType, aSpriteNum)
  case aSpriteType of
    #graphic:
      return getPos([#bitmap, #flash, #picture, #text, #vectorShape],
        sprite(aSpriteNum).member.type) <> 0
    #script:

```

```

    return FALSE
  end case
end isOKToAttach

on getPropertyDescriptionList me

  if not the currentSpriteNum then exit
end getPropertyDescriptionList

```

1. Draggable: kéo rê đối tượng

```

-- Draggable
-- Click and drag sprite to new location
-- 16 Nov DP - incorporated JN fix for compatibility with Draw
Connector
-- v1 - 7 October 1998 by Darrel Plant
-- Modified 1 February, 1999 by A.M. Kelsey
-- 7 January 2000 Added isOkToAttach J.Powers

on getBehaviorDescription me
  return \
    "DRAGGABLE" & RETURN & RETURN & \
    "Makes sprite respond to clicking and dragging the mouse button,
similar to setting the draggable property of the sprite. " & \
    "Sprite movement may constrained to the stage area." & RETURN &
RETURN & \
    "PERMITTED MEMBER TYPES:" & RETURN & \
    "Graphic members" & RETURN & RETURN & \
    "If using with digital video or other sprites in which Direct to
Stage is an option, disable Direct to Stage for best results." &
RETURN & RETURN & \
    "PARAMETERS:" & RETURN & \
    "* Constrain to stage?"
end getBehaviorDescription

on getBehaviorTooltip me
  return \
    "Makes a sprite draggable. " & \
    "The sprite will move along with the cursor while the mouse
button is held down."
end getBehaviorTooltip

-- PROPERTIES --

property pSprite -- sprite object
property pLocOffset -- offset of click from sprite loc
property pActive -- activity flag
-- author-defined properties
property pConstrained -- constrain to stage flag

-- EVENT HANDLERS --

```

```

on beginSprite me
    -- determine sprite object reference
    pSprite = sprite (me.spriteNum)
    vMember = pSprite.member
    case vMember.type of
    #animGIF, #flash, #QuickTimeMedia, #digitalvideo, #vectorShape:
    if vMember.directtostage then
        alert "Sprite" && pSprite.spriteNum & \
            ": Direct To Stage media may cause" && \
            "playback problems with the 'Moveable Sprite' behavior."
        end if
    end case
    -- set activity flag
    pActive = FALSE
end beginSprite

on mouseUp me
    -- turn off activity flag when mouse is released
    mDrag FALSE
end mouseUp

on mouseDown me
    -- turn on activity flag when mouse is clicked
    mDrag TRUE
end mouseUp

on prepareFrame me
    -- if mouse has somehow moved off of sprite and been released
    -- turn off activity flag
    if the mouseUp then mDrag FALSE
    -- if active, move sprite to follow cursor
    if pActive then
        mDragSprite me
    end if
end prepareFrame

-- CUSTOM HANDLERS --

on mDrag vActive
    -- set activity flag
    pActive = vActive
    -- find distance from sprite loc to mouse click
    pLocOffset = pSprite.loc - the mouseLoc
    -- inform other behaviors of current state
    sendAllSprites (#Active_Sprite, the currentSpriteNum * pActive)
end mDrag

on mDragSprite
    -- move sprite to cursor location offset by difference between
    -- original click and sprite loc
    pSprite.loc = the mouseLoc + pLocOffset

```

```

if pConstrained then
vLeftDiff = pSprite.rect.left
if vLeftDiff < 0 then
pSprite.locH = pSprite.locH - vLeftDiff
end if
vRightDiff = pSprite.rect.right - the stage.sourcerect.width
if vRightDiff > 0 then
pSprite.locH = pSprite.locH - vRightDiff
end if
vTopDiff = pSprite.rect.top
if vTopDiff < 0 then
pSprite.locV = pSprite.locV - vTopDiff
end if
vBottomDiff = pSprite.rect.bottom - the stage.sourcerect.height
if vBottomDiff > 0 then
pSprite.locV = pSprite.locV - vBottomDiff
end if
end if
end mDragSprite

```

```
-- INTER-BEHAVIOR COMMUNICATION
```

```

on Forwarded_mouseDown me
-- turn on activity flag when mouse click is initially
-- intercepted by another behavior and forwarded
if not pActive then
mDrag TRUE
end if
return the currentSpriteNum
end Forwarded_mouseDown

```

```
-- AUTHOR-DEFINED PARAMETERS --
```

```

on isOKToAttach (me, aSpriteType, aSpriteNum)
return aSpriteType = #Graphic
end

on getPropertyDescriptionList
vPDList = [:]
setaProp vPDList, #pConstrained, [#comment: "Constrain to
stage", \
#format: #boolean, #default: TRUE]
return vPDList
end getPropertyDescriptionList

```

2. end: màn hình kết thúc

```

property pchannel
on mousedown me

```

```

sprite(pchannel).movietime=member(sprite(pchannel).member).durati
on
  sprite(pchannel).movieRate=1
end

on getPropertyDescriptionList me
  set mylist=[]

  addProp mylist, # pChannel, [#default: 2, #format: #integer,
#range: [#min: 1, #max:70], #comment: "member channel"]
  return mylist
end

```

3. *Hold on Current Frame: dừng tại vị trí khung hình hiện tại*

-- DESCRIPTION --

```

on getBehaviorDescription me
  return \
    "HOLD ON CURRENT FRAME" & RETURN & RETURN & \
    "Drop this behavior into the Script Channel of the Score or onto
the Stage in order to keep the playback head in the current
frame." & RETURN & RETURN & \
    "PARAMETERS: None"
end getBehaviorDescription

```

```

on getBehaviorTooltip me
  return \
    "Frame behavior. " & \
    "Holds the playback head still."
end getBehaviorTooltip

```

-- HISTORY --

-- 3 November, written for the D7 Behaviors Palette by James
Newton
-- 5 January 2000: updated to D8 <km>

```

on exitFrame me
  go the frame
end exitFrame

```

```

on isOKToAttach (me, aSpriteType, aSpriteNum)

  tIsOk = 0

```

```

if aSpriteType = #script then
tIsOK = 1
end if

return(tIsOK)
end on

```

4. *Jump to Marker Button: nhảy tới vị trí đánh dấu*

```
-- DESCRIPTION --
```

```

on getBehaviorDescription me
return \
"JUMP TO MARKER BUTTON" & RETURN & RETURN & \
"A click on the sprite makes the playback head jump to a chosen
marker in the same movie." & \
"You can choose between the previous marker, the next marker, or
any named marker in the movie." & RETURN & RETURN & \
"If more than one marker has the same name as the chosen marker,
the playback head will jump to the first marker with that name in
the movie." & \
"If you need to use duplicate names, try adding a different
number of spaces at the end of each, so that they appear the same
but are in fact different." & RETURN & RETURN & \
"This behavior is designed for use with the Jump Back Button. "
& \
"You can set it to record the frame number of the current marker
so that the Jump Back Button can subsequently return." & \
"Note: This feature relies on a global variable:
gNavigationBarList." & RETURN & RETURN & \
"Use the Jump to Movie Button behavior to jump to a marker in a
different movie." & RETURN & RETURN & \
"PERMITTED MEMBER TYPES:" & RETURN & \
"Graphic members" & RETURN & RETURN & \
"PARAMETERS:" & RETURN & \
"* Jump to marker in this movie" & RETURN & \
"* 'Go to' or 'Play and return'?" & RETURN & \
"* Remember current marker for Back button" & RETURN & RETURN & \
\
"Select Remember Current Marker for Back Button to ensure that
the behavior 'remembers' which markers have already been
visited." & RETURN & RETURN & \
"Use the associated Jump Back Button behavior on a separate
sprite to return to visited markers in reverse order." & RETURN &
RETURN & \
"ASSOCIATED BEHAVIORS:" & RETURN & \
"* Jump to Movie button" & RETURN & \
"* Jump Back Button" & RETURN & \
"* Jump Forward Button" & RETURN & \
"* Push Button (to alter rollover / mouseDown states)"
end getBehaviorDescription

```

```

on getBehaviorTooltip me
    return \
        "Use with graphic members." & RETURN & RETURN & \
        "When you drop this behavior on a sprite, you can choose which
marker in the current movie to jump to on mouseUp. " & \
        "You can also choose to jump to the next or the previous
marker." & RETURN & RETURN & \
        "Use this behavior with the Jump Back Button to allow the user
to return to visited sequences."
end getBehaviorTooltip

```

```
-- NOTES FOR DEVELOPERS --
```

```

-- The main handler is Jump, which is triggered by mouseUp. The
Initialize
-- handler runs a series of error-checks. These will alert
designers
-- in any of the following cases:
-- * if the chosen marker has been deleted or renamed
-- * if more than one marker has the same name as the chosen
marker
-- * if the global gNavigationButtonList does not have a valid
structure

```

```
-- USE OF 'BACK/FORWARD BUTTON' BEHAVIORS
```

```

--
-- This behavior is designed to be use in collaboration with
'Jump Back' and
-- 'Jump Forward' buttons so that users can retrace their steps.
--
-- Information about what markers have already been visited is
stored in a
-- global variable: gNavigationButtonList. As a global, this
variable is
-- available to all navigation behaviors, on any sprite in any
movie.

```

```
-- gNavigationButtonList has the following structure:
```

```
-- [#stack [...], #index: [...], #forward: [...]]
```

```
-- Each of the subLists has the structure:
```

```
-- [[#frame: <integer>, #movie: <movieName>], [...]]
```

```
--
-- #index is not currently exploited. It provides a list of all
markers
```

```
-- visited, in the order of their first visit. It could be used
to create
```

```
-- a pop-up menu, or to check if a player had visited a
particular scene.
```

```

--
-- Information concerning the most recently visited marker is
stored in the

```

```

-- last item of the #stack list. This item is copied to a local
variable,
-- then removed from the #stack. Information concerning the
current marker
-- is appended to the #forward list, so that users can
subsequently retrace
-- their steps.

-- HISTORY --

-- 10 September 1998: written for the D7 Behaviors Palette by
James Newton
-- 22 October 1998: "play"/"go to" added, error-checking eased to
allow
-- authors to use the behavior before creating markers.
-- 5 January 2000: updated to D8 <Kraig Mentor>
-- 14 February 2000: Added resolve handler. Added markerList
conversion to
-- GDPL hander.

-- PROPERTIES --

-- author-defined parameters:
property myMarker
property myJumpMode
property myReturn

global gNavigationButtonList -- Only declared if it is to be used

-- EVENT HANDLERS --

on beginSprite me
    resolve(me)
    Initialize me
end beginSprite

on mouseUp me
    Jump me
end prepareFrame

-- CUSTOM HANDLERS --

on resolve me
    --This handler is used to convert non-English languages to Lingo
terms.
    case myMarker of
        "next":

```

```

myMarker = #next
"loop":
myMarker = #loop
"previous":
myMarker = #previous
end case
end on

on Initialize me -- sent BY beginSprite
-- Error checking: Marker name
if not getPos ([#next, #previous, #loop], myMarker) then
-- Does myMarker still exist?
theMarkerList = RETURN&the labelList
markerExists = offset (RETURN&myMarker&RETURN, theMarkerList)
if not markerExists then
ErrorAlert me, #markerMissing, myMarker
else
-- Is myMarker still unique?
delete char 1 to markerExists of theMarkerList
if offset (RETURN&myMarker&RETURN, theMarkerList) then
ErrorAlert me, #runtime_DuplicateMarkers, myMarker
end if
end if
end if

-- Error checking: Global variable
if myReturn then
global gNavigationButtonList -- Only declared if it is to be
used

if voidP (gNavigationButtonList) then
-- Initialize gNavigationButtonList
gNavigationButtonList = [#stack: [], #forward: [], #index: []]
else -- Is it the right global?
if gNavigationButtonList.ilkk <> #propList then
ErrorAlert (me, #invalidGlobal, gNavigationButtonList)
else if not gNavigationButtonList.findPos(#stack) then
ErrorAlert (me, #invalidGlobal, gNavigationButtonList)
else if not gNavigationButtonList.findPos(#index) then
ErrorAlert (me, #invalidGlobal, gNavigationButtonList)
else if not gNavigationButtonList.findPos(#forward) then
ErrorAlert (me, #invalidGlobal, gNavigationButtonList)
end if
end if
end if
-- End of error checking
end Initialize

on Jump me -- sent by mouseUp
case myMarker of

```

```

    #previous:targetIsNewMarker = (marker (-the maxInteger / 2) <>
marker (0))
    #loop: targetIsNewMarker = FALSE
    #next: targetIsNewMarker = (marker (0) <> marker (the maxInteger
/ 2))
    otherwise
    targetIsNewMarker = (marker (0) <> marker (myMarker))
    end case

    if myReturn and targetIsNewMarker then
    -- Remember where to come back to
    global gNavigationButtonList -- Only declared if it is to be
used

    currentMarker = [#frame: marker (0), #movie: the movieName]
    -- Add currentMarker to #stack
    gNavigationButtonList.stack.append(currentMarker)
    -- Check if currentMarker is already in #index
    if not gNavigationButtonList.index.getPos(currentMarker) then
    gNavigationButtonList.index.append(currentMarker)
    end if
    -- Empty #forward list
    gNavigationButtonList.forward = []
    end if

    -- Go to frame indicated by myMarker
    if myJumpMode = "Go to" then
    case myMarker of
    #previous: go previous
    #loop: go loop
    #next: go next
    otherwise
    go marker (myMarker)
    end case
    else -- play => play done
    case myMarker of
    #previous: play marker (-1)
    #loop: play marker (0)
    #next: play marker (1)
    otherwise:
    play myMarker
    end case
    end if
    end Jump

on substituteStrings(me, parentString, childStringList) -----
-----
    -- Sent by errorAlert
    --
    -- * Modifies parentString so that the strings which appear as
    -- properties in childStringList are replaced by the values
    -- associated with those properties.

```

```

--
-- <childStringList> has the format ["^1": "replacement string"]
-----

i = childStringList.count()
repeat while i
tempString = ""
dummyString = childStringList.getPropAt(i)
replacement = childStringList[i]
lengthAdjust = dummyString.char.count - 1
repeat while TRUE
position = offset(dummyString, parentString)
if not position then
parentString = tempString&parentString
exit repeat
else
if position <> 1 then
tempString = tempString&parentString.char[1..position - 1]
end if
tempString = tempString&replacement
delete parentString.char[1..position + lengthAdjust]
end if
end repeat
i = i - 1
end repeat

return parentString
end substituteStrings

-- ERROR CHECKING --

on ErrorAlert me, theError, data
-- Send by getPropertyDescriptionList, initialize
-- Determine the behavior's name
behaviorName = string(me)
delete word 1 of behaviorName
delete the last word of behaviorName
delete the last word of behaviorName
-- Convert #data to useful value
case data.ilk of
#void: data = "<void>"
#symbol: data = "#"&data
end case

case theError of

#getPDL_DuplicateMarkers:
tError1 = "Two or more markers in this movie have the same name.
" & \
" This could lead to " & RETURN & \

```

```

"confusion if you use this behavior to go to a named marker."
tError2 = "Duplicate marker(s) = ^0"
tError2 = substituteStrings(me, tError2, ["^0":data])
alert(tError1 & RETURN & RETURN & tError2)

#markerMissing:
if the runMode = "Author" then
tError1 = "BEHAVIOR ERROR: Frame ^0, Sprite ^1"
tError1 = substituteStrings(me, tError1, ["^0":the frame, "^1":
the currentSpriteNum])
tError2 = "Behavior ^0"
tError2 = substituteStrings(me, tError2, ["^0": behaviorName])
tError3 = "The chosen marker no longer exists." & RETURN & \
"Choose a valid marker in the Behavior Parmeters dialog."
tError4 = "Current marker = ^0"
tError4 = substituteStrings(me, tError4, ["^0": data])
alert(tError1 & RETURN & RETURN & tError2 & RETURN & RETURN &
tError3 & RETURN & RETURN & tError4)
end if
abort

#runtime_DuplicateMarkers:
if the runMode = "Author" then
tError1 = "BEHAVIOR ERROR: Frame ^0, Sprite ^1"
tError1 = substituteStrings(me, tError1, ["^0":the frame, "^1":
the currentSpriteNum])
tError2 = "Behavior ^0"
tError2 = substituteStrings(me, tError2, ["^0": behaviorName])
tError3 = "The chosen marker does not have a unique name. " & \
" The playback head will jump to the first marker with this
name:"
tError4 = "Duplicate marker name = ^0"
tError4 = substituteStrings(me, tError4, ["^0": data])
alert(tError1 & RETURN & RETURN & tError2 & RETURN & RETURN &
tError3 & RETURN & RETURN & tError4)
end if

#invalidGlobal:
tError1 = "BEHAVIOR ERROR: Frame ^0, Sprite ^1"
tError1 = substituteStrings(me, tError1, ["^0":the frame, "^1":
the currentSpriteNum])
tError2 = "Behavior ^0"
tError2 = substituteStrings(me, tError2, ["^0": behaviorName])
tError3 = "[#stack [...], #index: [...], #forward: [...]]"
tError4 = "Current value = ^0"
tError4 = substituteStrings(me, tError4, ["^0": data])
alert(tError1 & RETURN & RETURN & tError2 & RETURN & RETURN &
tError3 & RETURN & RETURN & tError4)
halt
end case
end ErrorAlert

```

```

on isOKToAttach (me, aSpriteType, aSpriteNum)

    tIsOK = 0
    if aSpriteType = #graphic then
        tIsOK = 1
    end if

    return(tIsOK)
end on

-- AUTHOR-DEFINED PARAMETERS --

on getPropertyDescriptionList me

    currentMember = sprite(the currentSpriteNum).member

    markersList = GetMarkers (me)
    if markersList.count() = 2 then -- [[<markers>], [<duplicates>]]
        ErrorAlert (me, #getPDL_DuplicateMarkers, markersList[2])
    end if

    tLabelList = []
    tLabelList.add("next")
    tLabelList.add("loop")
    tLabelList.add("previous")

    repeat with i = 1 to count(the markerList)
        tLabelList.add((the markerList).getProp((the
markerList).getPropAt(i)))
    end repeat

    return \
[ \
    #myMarker: \
    [ \
        #comment: "On mouseUp, jump to marker", \
        #format: #string, \
        #range: tLabelList,\
        #default: "next"\
    ], \
    #myJumpMode: \
    [ \
        #comment: "Jump Mode", \
        #format: #string, \
        #range: ["Go to", "Play and Return"], \
        #default: "Go to" \
    ], \
    #myReturn: \
    [ \

```

```

    #comment: "Remember current marker for Back button?", \
    #format: #boolean, \
    #default: TRUE \
  ] \
]
end getPropertyDescriptionList

on GetMarkers me -- Sent by getPropertyDescriptionList
  localMarkerList = []
  duplicatesList = []
  markerString = the labelList
  delete the last char of markerString
  markerCount = the number of lines of markerString
  repeat with i = 1 to markerCount
    theMarker = markerString.line [i]
    if localMarkerList.getPos(theMarker) then
      -- Duplicate marker name
      if not duplicatesList.getPos(theMarker) then
        duplicatesList.append(theMarker)
      end if
    else
      localMarkerList.append(theMarker)
    end if
  end repeat
  if duplicatesList.count() then
    return [localMarkerList, duplicatesList]
  else
    return [localMarkerList]
  end if
end GetMarkers

```

5. Multi-State Button: điều khiển cùng lúc nhiều trạng thái

```

-- DESCRIPTION --

on getBehaviorDescription me
  return \
    "MULTISTATE TOGGLE BUTTON" & RETURN & RETURN & \
    "This behavior toggles the sprite it is attached to between two  

states: OFF and ON. " & \
    "In each state, the member of the sprite is set according to the  

position of the mouse (elsewhere, rollover, mouseDown)." & RETURN  

& \
    "The behavior returns the current state of the button in  

response to a #ToggleButton_State call." & RETURN & RETURN & \
    "RADIO BUTTON GROUP" & RETURN & \
    "To create a group of radio buttons, give each button in the  

same group the same ID. " & \
    "Switching one button in the group on will switch all others  

off." & RETURN & RETURN & \
    "PERMITTED MEMBER TYPES" & RETURN & \
    "Graphic" & RETURN & RETURN & \

```

```

"PARAMETERS:" & RETURN & \
"-- OFF state --" & RETURN & \
"* Standard member (when mouse is elsewhere)" & RETURN & \
"* Rollover member" & RETURN & \
"* MouseDown member" & RETURN & \
"-- ON state --" & RETURN & \
"* Standard member" & RETURN & \
"* Rollover member" & RETURN & \
"* MouseDown member" & RETURN & \
"-- COMMAND --" & RETURN & \
"* Sent when the button is switched ON" & RETURN & \
"* Sent when the button is switched OFF" & RETURN & \
"-- Optional --" & RETURN & \
"* Toggle group ID (to create a group of radio buttons)" &
RETURN & RETURN & \
"If members are placed consecutively in the cast in this order
then default values can be used to create the button."
end getBehaviorDescription

on getBehaviorTooltip me
return \
"Create an ON/OFF button which reacts to rollovers and clicks. "
& \
"Use several such buttons together as a radio button group."
end getBehaviorTooltip

-- NOTES FOR DEVELOPERS --
--
-- This behavior communicates extensively with other sprites.
Since there
-- may be up to 1000 sprites, it is important to use
sendAllSprites with
-- prudence. Indeed, this behavior only uses sendAllSprites once,
in the
-- Initialize handler.
--
-- ToggleButton_GroupRollCall
-- The purpose of the call is to identify all other members of
the radio
-- button group. The behaviors that field the call add their
object reference
-- to a list, ourGroupList. Each behavior stores a pointer to
this list.
-- Subsequent communications are of the form:
--
-- call (#customMessage, ourGroupList, additionalParameters)
--
-- This limits the messaging process to only those sprites which
need to know.
--

```

```

-- ToggleButton_MouseDownList
-- Lists can be used to make information available to other
sprites without
-- sending any messages. All behaviors in a group also share a
pointer to an
-- ourMouseDown list. Whenever the user clicks on one of the
members, the
-- unique item in this list is set to TRUE. When the mouse is
released, it
-- is set to FALSE. All buttons in the group know instantaneously
if any
-- member of the group has been clicked. This means that if the
mouse is
-- clicked on one button in the group, other group members will
switch to
-- their mouseDown state as the mouse is draggged over them
--
-- ToggleButton_Rollover
-- When the mouse is released, the clickOn sprite asks each other
group
-- member in turn if the mouse is currently over it. If none
respond, then
-- ourMouseDown is set to [FALSE]. If any buttons respond, then
the
-- mouseUp event sent to the topmost button will activate its
Toggle handler.
--
-- ToggleButton_Off
-- When a button is toggled on, it instructs all others in the
group to switch
-- themselves off.
--
-- ToggleButton_State, ToggleButton_ActiveButton
-- These two calls are not used in the behavior itself. they are
included
-- to let you know the state of a particular button, or to know
which button
-- in a given group is currently ON.
--
-- If you have only one button, then you can use the
ToggleButton_State
-- call with no parameters. If there are several buttons, then
you can either
-- send an empty list out to a given group, and receive a list of
the state of
-- each button in reply.
--
-- ToggleButton_ActiveButton should be sent to a particular
group. You
-- must include an empty property list in your call, and will
receive in reply
-- a list of the form: [#sprite: <spriteNum>, #behavior: <object
reference>]

```

```

--
--
--
-- HISTORY --
--
-- 11 September 1998, written for the D7 Behaviors Palette by
James Newton
-- 5 January 2000: updated to D8 <km>


-- PROPERTIES --

property spriteNum
property mySprite
-- author-defined parameters
property myOffMember
property myOffOverMember
property myOffDownMember
property myOffCommand
property myOnMember
property myOnOverMember
property myOnDownMember
property myOnCommand
-- internal properties
property theMouseWasUp
property myRollover
property myState -- TRUE | FALSE: max one button in group TRUE at
one time
-- shared properties
property ourID -- string common to all buttons in a group
property ourGroupList -- list of behaviors in the group
property ourMouseDown -- list indicating if the clickOn is in the
group


-- EVENT HANDLERS --

on beginSprite me
  Initialize me
end beginSprite

on endSprite(me)
  ourGroupList.deleteOne(me)
end

on prepareFrame me
  CheckForRollover me
end prepareFrame

```

```

on mouseDown me
  ClickOn me
end mouseDown

on mouseUp me
  if ourMouseDown[1] then Toggle me
end mouseUp

on mouseUpOutside me
  CheckGroupForClick me
end mouseUpOutside

-- CUSTOM HANDLERS --

on Initialize me -- sent by beginSprite
  mySprite = sprite(me.spriteNum)
  ourGroupList = []

  -- Insurance: properties are indeed #members
  myOffMember = member (myOffMember)
  myOffOverMember = member (myOffOverMember)
  myOffDownMember = member (myOffDownMember)
  myOnMember = member (myOnMember)
  myOnOverMember = member (myOnOverMember)
  myOnDownMember = member (myOnDownMember)

  sendAllSprites (#ToggleButton_GroupRollCall, ourID,
ourGroupList)
  call (#ToggleButton_MouseDownList, ourGroupList, [FALSE])
end Initialize

on CheckForRollover me -- sent by prepareFrame
  mouseOverMe = (the rollover = me.spriteNum)
  if myRollover = mouseOverMe then
    if theMouseWasUp = the mouseUp then
      exit -- Nothing has changed

    else
      theMouseWasUp = the mouseUp
      if mouseOverMe then
        if the mouseUp then
          -- Mouse was clicked elsewhere then dragged and released over
button
          case myState of
            TRUE: mySprite.member = myOnOverMember
            FALSE: mySprite.member = myOffOverMember

```

```

end case
end if
end if
end if
else

set myRollover to mouseOverMe
if ourMouseDown[1] then
if myRollover then
case myState of
TRUE: mySprite.member = myOnDownMember
FALSE: mySprite.member = myOffDownMember
end case
else
-- Indicate that mouseUpOutside will have no effect
case myState of
TRUE: mySprite.member = myOnMember
FALSE: mySprite.member = myOffMember
end case
end if
else
if not the mouseDown and myRollover then
case myState of
TRUE: mySprite.member = myOnOverMember
FALSE: mySprite.member = myOffOverMember
end case
else
-- No reaction if mouse was clicked elsewhere and dragged to
button
case myState of
TRUE: mySprite.member = myOnMember
FALSE: mySprite.member = myOffMember
end case
end if
end if
end if
end CheckForRollover

on ClickOn me -- sent by mouseDown, CheckForRollover
ourMouseDown[1] = TRUE
case myState of
TRUE: mySprite.member = myOnDownMember
FALSE: mySprite.member = myOffDownMember
end case
end ClickOn

on Toggle me, whichState -- sent by mouseUp, ToggleButton_Off
if voidP (whichState) then
myState = not myState

```

```

else
myState = whichState
end if
ourMouseDown[1] = FALSE
theMouseWasUp = TRUE
case myState of
TRUE:
mySprite.member = myOnMember
updateStage
do myOnCommand
FALSE:
mySprite.member = myOffMember
updateStage
do myOffCommand
end case
if ourGroupList.count() then
if myState then
call (#ToggleButton_Off, ourGroupList, me)
end if
end if
end Toggle

on CheckGroupForClick me -- sent by mouseUpOutside
groupRollover = call (#ToggleButton_Rollover, ourGroupList, [])
if not groupRollover.count() then
ourMouseDown[1] = FALSE
end if
end Disactivate

-- PUBLIC METHODS (responses to #sendSprite, #sendAllSprites,
#call) --

on ToggleButton_GroupRollCall me, groupID, groupList
-- sent by each new button that joins the group
if groupID = ourID then
ourGroupList = groupList
ourGroupList.append(me)
end if

```

```
i = childStringList.count()
repeat while i
    tempString = ""
    dummyString = childStringList.getPropAt(i)
    replacement = childStringList[i]
    lengthAdjust = dummyString.char.count - 1
    repeat while TRUE
        position = offset(dummyString, parentString)
        if not position then
            parentString = tempString&parentString
            exit repeat
        else
            if position <> 1 then
                tempString = tempString&parentString.char[1..position - 1]
            end if
            tempString = tempString&replacement
            delete parentString.char[1..position + lengthAdjust]
        end if
    end repeat
    i = i - 1
end repeat

return parentString
end substituteStrings

-- AUTHOR-DEFINED PARAMETERS --

on isOKToAttach (me, aSpriteType)
    return aSpriteType = #graphic
end isOKToAttach

on getPropertyDescriptionList me

    -- try to find a sprite which has the 'Display Text'
    behavior attached
    displaySprite = GetDisplaySprite (me)

    return \
    [ \
        #myString: \
        [ \
            #comment: "Text of tool tip", \
            #format: #string, \
            #default: "Insert your single-line tool tip here" \
        ], \
        #myDelay: \
        [ \
            #comment: "Pause before showing tool tip (ticks)", \
            #format: #integer, \
            #range: [#min: 0, #max: 120], \
            #default: 30 \
        ], \
        #myHideFlag: \
        [ \
            #comment: "Hide tool tip if user clicks on sprite?", \
            #format: #boolean, \
            #default: TRUE \
        ], \
    ]
```

```
#myPosition: \
[ \
#comment: "Tool tip position relative to sprite (see
notes)", \
#format: #string, \
#range: \
[ \
"centered above", \
"at topLeft", \
"at topRight", \
"centered", \
"at bottomLeft", \
"at bottomRight", \
"centered below", \
"at regPoint", \
"under the mouse" \
], \
#default: "centered" \
], \
#myDisplaySprite: \
[ \
#comment: "Use which sprite to display tooltip?", \
#format: #integer, \
#range: [#min: 1, #max: the lastChannel], \
#default: displaySprite \
] \
]
end getPropertyDescriptionList
```

1. Typewriter Effect: hiệu ứng khi di chuyển chuột vào dòng văn bản

```
-- Typewriter Effect
-- Uses a text or field cast member to simulate text typed on
screen.
-- The amount of time between each character can be set,
-- and a sound can be played for each character. Author sets
up full
-- text to be typed in field, adds behavior and lets it rip.
-- v1 - 3 October 1998 by Darrel Plant
--
-- Modified 10 January, 2000 by Tom Higgins: added the
isOKToAttach
-- event handler, removed redundant error checking.
```

```
on getBehaviorDescription me
return \
"TYPEWRITER EFFECT" & RETURN & RETURN & \
"A field or text sprite with this behavior attached displays
as if it was being typed onto the screen. " & \
"To use, place a field or text cast member in place on the
stage; and type in the message you want to appear; then
attach this behavior. " & \
"The type can begin to display automatically in the first
frame the sprite appears in, or it can be set to wait for an
mActive message. " & \
```

```
"You can set the amount of time to wait between characters,
and play a sound for each letter." & RETURN & RETURN & \
"PARAMETERS:" & RETURN & \
" - Autotype when sprite appears" & RETURN & \
" - Time between characters" & RETURN & \
" - Sound to play when character appears" & RETURN & \
" - Sound channel to use" & RETURN & RETURN & \
"PERMITTED TYPES:" & RETURN & \
" - Text or Fields"
end getBehaviorDescription

on getBehaviorTooltip me
    return \
    "Types text onto screen into 'visible' or 'invisible' text
    field. " & \
    "An accompanying sound can be assigned to play as text
    appears."
end getBehaviorTooltip

-- PROPERTIES --

property pSprite -- sprite object reference
property pMember -- member object reference
property pText -- text to type into field or text cast member
property pChars -- length of text
property pIndex -- number of character being displayed
property pStartTime -- last time character was displayed
-- author-defined properties
property pActive -- autostart property for behavior
property pPeriodBase -- number of seconds between characters
property pPeriod -- pPeriodBase converted to milliseconds
property pSoundChannel -- also used as flag to determine if
sound is played
property pSound -- sound to play for character typing

-- EVENT HANDLERS --

on beginSprite me
    mInitialize me
end beginSprite

on endSprite me
    pMember.text = pText
end

on prepareFrame me
    mType me
end prepareFrame

-- CUSTOM HANDLERS --

on mInitialize me
    -- called by beginSprite
    -- determine sprite reference
    pSprite = sprite (me.spriteNum)
    -- determine cast member reference
    pMember = pSprite.member
```

```
-- set framing for text or field member to #fixed so it
doesn't change size
pMember.boxType = #fixed
-- extract text from text or field cast member
pText = pMember.text
-- find number of characters to display
pChars = pText.length
-- show alert if field or text member is empty
if pChars < 1 then
vError = mErrorAlert (me, #noText)
end if
-- put space in field or text member to preserve formatting
pMember.text = " "
-- index keeps track of which character is currently being
displayed
pIndex = 0
-- convert seconds set by user into milliseconds for
behavior
pPeriod = integer (pPeriodBase * 1000)
-- timer for counting periods between characters, set for
immediate
-- displ of first character
pStartTime = the milliseconds - pPeriod
-- check to see if a valid sound cast member is available
-- only check if a sound channel is selected
if pSoundChannel then
-- flag to determine if error is detected
vError = FALSE
-- sound channel selected, no sound selected
if pSound = #none then
vError = TRUE
else
-- sound channel selected member chosen is not sound
if member (pSound).type <> #sound then
vError = TRUE
end if
end if
-- sound error detected
if vError then
-- set sound channel to 0
pSoundChannel = 0
-- display no sound error
vError = mErrorAlert (me, #noSound)
end if
end if
end mInitialize

on mType me
-- called from prepareFrame handler
-- only evaluated if behavior is active
if pActive then
-- if index value is larger than the number of characters,
-- all characters should have been shown
if pIndex >= pChars then
-- behavior should no longer be active
pActive = FALSE
else
-- get current millisecond value
```

```
vMillis = the milliseconds
-- determine if enough time has elapsed since last character
was shown
if (vMillis - pStartTime) > pPeriod then
  -- increment index value
  pIndex = pIndex + 1
  if pIndex = 1 then
    -- because a space is used in the field or text member to
preserve
    -- formatting, the first character needs to replace the
space
    pMember.text = pText.char[1]
  else
    -- otherwise each character in the text is concatenated with
the
    -- text or field member
    pMember.text = pMember.text & pText.char[pIndex]
  end if
  -- check for sound flag, if no channel is chosen, nothing
happens
  if pSoundChannel then
    -- kill sound in channel
    puppetsound pSoundChannel, 0
    -- play sound in specified sound channel
    puppetSound pSoundChannel, pSound
  end if
  -- update timer value
  pStartTime = the milliseconds
end if
end if
end if
end mType

on mErrorAlert me, vError, vData
  -- based on James Newton's error checking procedure
  -- determine name of behavior
  vBehaviorname = string (me)
  delete word 1 of vBehaviorName
  delete the last word of vBehaviorName
  delete the last word of vBehaviorName
  -- convert supporting data
  case vData.ilk of
  #void: vData = "<void>"
  #symbol: vData = "#" & vData
  end case
  case vError of
  -- deal with individual error types
  #noText:
    alert "CANCEL: Sprite MUST contain formatted text to display
correctly"
  #noSound:
    alert "CANCEL: Sound channel should be set to 0 if no sound
is selected"
  end case
end mErrorAlert

-- PUBLIC METHODS --
--Example:
```

```
--
--on exitFrame
-- sendSprite(whichSprite, #mActive)
--end
--
--replace "whichSprite" with the number of the sprite you
wish to start typing
on mActive me
  set pActive = TRUE
end mActive

-- AUTHOR-DEFINED PARAMETERS --

on isOKToAttach (me, aSpriteType, aSpriteNum)
  case aSpriteType of
    #graphic:
      return getPos([#field, #text], \
        sprite(aSpriteNum).member.type) <> 0
    #script:
      return FALSE
  end case
end isOKToAttach

on getPropertyDescriptionList me
  vPDLList = [:]
  setaProp vPDLList, #pActive, \
    [#comment: "Autostart (alternative is wait for mActive
message)", \
    #format: #boolean, #default: TRUE]
  setaProp vPDLList, #pPeriodBase, \
    [#comment: "Time to wait between typed characters
(seconds)", \
    #format: #float, #default: 0.250, #range: [#min: 0.001,
#max: 2.000]]
  setaProp vPDLList, #pSound, \
    [#comment: "Sound for typed characters", #format: #sound,
#default: #none]
  setaProp vPDLList, #pSoundChannel, \
    [#comment: "Channel for sound (0 for no sound)", \
    #format: #integer, #default: 0, #range: [0, 1, 2, 3, 4, 5,
6, 7, 8]]
  return vPDLList
end getPropertyDescriptionList
```

2. *xhchau*: điều khiển đối tượng chậu nước

```
global n2,i,r
on mouseup me
  if ( sprite(17).visible=true) then
    sprite(17).visible=false
    n2=1
  else
    sprite(17).visible= true
    if sprite(12).visible=true then sprite(18).visible=true
    n2=4/(3).float
  end if
```

```
if n2=4/(3).float then
if i<>0 then
sinr=sin(i*PI/180)*3/4
cosr=sqrt(1-sinr*sinr).float
r=(atan(sinr/cosr)*180/PI ).integer
else
r=0
end if
else
r=i
end if
sprite(13).rotation=90-r

end
```

3. *xhden: điều khiển đối tượng nguồn sáng*

```
on mouseup me
if sprite(15).visible=false then
sprite(15).visible=true
sprite(16).visible= true
if sprite(12).visible= true then
sprite(12).visible= false
sprite(13).visible= false
end if

else
repeat with j=11 to 16
sprite(j).visible=false
end repeat
end if
end
```

4. *do do: điều khiển thước đo độ*

```
global i2, r2
on exitframe me
i2=i2.integer
r2=r2.integer
if (sprite(me.spritenum).loc=sprite(12).loc) and
(abs(sprite(me.spritenum).rotation>270) and
(abs(sprite(me.spritenum).rotation) mod 360<2)) then
sprite(58).visible=true
member("hien i").text=""
member("hien i").text="i=" & string(i2)
else
sprite(58).visible=false
end if

if(sprite(me.spritenum).loc=sprite(12).loc)and
(abs(sprite(me.spritenum).rotation<270) and
(abs(sprite(me.spritenum).rotation) mod 180<2)) then
sprite(59).visible=true
member("hien r").text=""
member("hien r").text="r=" & string(r2)
```

```
else
sprite(59).visible=false
end if

end
```

5. *tro ve sau huong dan: trở về sau hướng dẫn*

```
global currfr
on mouseup me
go to frame currfr
end
```

6. *Move, Rotate and Scale: di chuyển, quay và cân bằng đối tượng*

-- DESCRIPTION --

```
on getBehaviorDescription me
return \
"MOVE, ROTATE AND SCALE" & RETURN & RETURN & \
"Use customizable modifier keys to alter the way a sprite
reacts to the mouse. " & \
"If no modifier keys are pressed, clicking on the sprite and
dragging it moves the sprite. " & \
"If the 'rotate' modifier key is pressed, the sprite will
rotate about its registration point. " & \
"If the 'scale' modifier key is pressed, the sprite will
expand or contract according to the distance between the
mouse and the registration point. " & \
"If both modifier keys are pressed, rotation takes
priority." & RETURN & RETURN & \
"You can press and release the modifier keys as you drag the
sprite: its reaction will alter dynamically." & RETURN &
RETURN & \
"PERMITTED MEMBER TYPES:" & RETURN & \
"bitmap, Flash, vector shape" & RETURN & RETURN & \
"PARAMETERS:" & RETURN & \
"* Key to press to rotate sprite" & RETURN & \
"* Key to press to scale sprite" & RETURN & RETURN & \
"You can use any alphanumeric key or any of the following
words: Return, Tab, Space, Backspace, Command, Cmd, Shift,
ShiftLock, Option, Alt" & RETURN & RETURN & \
"PUBLIC METHODS:" & RETURN & \
"* Modify the action keys" & RETURN & \
"* Obtain behavior reference"
end getBehaviorDescription
```

```
on getBehaviorTooltip me
return \
"Use with bitmap, Flash and vector shape members." & RETURN
& RETURN & \
"Allows a sprite to be moved, rotated, or scaled by dragging
it with the mouse." & RETURN & RETURN & \
"The way the sprite reacts when the user drags it depends on
which custom action keys are being pressed. " & \
```

```
"The action key for 'rotate' has priority over the key for  
'scale'. " & \  
"With no action key pressed, the dragged sprite moves with  
the mouse."  
end getBehaviorTooltip
```

```
-- PROPERTIES --
```

```
property spriteNum  
-- error checking  
property getPDLError  
-- author-defined parameters  
property myRotateKey  
property myScaleKey  
-- internal properties  
property mySprite  
property myMouseDown  
property myPrevLoc  
property myPrevAngle  
property myPrevRect  
property myClickLoc  
property myDeltaLoc  
property myDeltaAngle  
property myDeltaLength  
property myAction
```

```
-- EVENT HANDLERS --
```

```
on beginSprite me  
  Initialize me  
end beginSprite me  
  
on mouseDown me  
  StartAction me  
end mouseDown  
  
on mouseUpOutside me  
  myMouseDown = FALSE  
end mouseUpOutside  
  
on mouseUp me  
  myMouseDown = FALSE  
end mouseUp  
  
on prepareFrame me  
  if myMouseDown then DoAction me  
end prepareFrame
```

```
-- CUSTOM HANDLERS --
```

```
on Initialize me -- sent by beginSprite  
  mySprite = sprite(me.spriteNum)
```

```
myMember = mySprite.member
memberType = myMember.type

case memberType of
#vectorShape: mySprite.scaleMode = #autoSize
end case

set myRotateKey to RationalizeKeys (me, myRotateKey)
set myScaleKey to RationalizeKeys (me, myScaleKey)
end Initialize

on RationalizeKeys me, theKey -- sent by Initialize,
_SetMessage
case theKey of
"Return": return 36
"TAB": return 48
"Space": return 49
"backspace": return 51
"Command", "Cmd": return 55
"Shift": return 56
"ShiftLock": return 57
"Option", "Alt": return 58
otherwise
return char 1 of theKey
end case
end RationalizeKeys

on StartAction me -- sent by mouseDown
myMouseDown = TRUE
myClickLoc = the clickLoc
myPrevLoc = mySprite.loc
myDeltaLoc = the clickLoc - myPrevLoc
end StartAction

on DoAction me -- sent by prepareFrame
if keyPressed (myRotateKey) then
Turn me
else if keyPressed (myScaleKey) then
Grow me
else
Drag me
end if
end DoAction

on Grow me -- sent by DoAction
if myAction <> #scale then
myAction = #scale
myPrevRect = mySprite.rect
myPrevLoc = mySprite.loc
myDeltaLoc = the mouseLoc - myPrevLoc
myDeltaLength = GetLength (me, myDeltaLoc)
end if

myDeltaLoc = the mouseLoc - myPrevLoc
```

```
deltaLength = GetLength (me, myDeltaLoc)
newRect = (myPrevRect * deltaLength) / myDeltaLength
mySprite.rect = newRect
mySprite.loc = myPrevLoc
end Grow
```

```
on Turn me -- sent by DoAction
  if myAction <> #rotate then
    myAction = #rotate
    myPrevAngle = mySprite.rotation
    myPrevLoc = mySprite.loc
    myDeltaLoc = the mouseLoc - myPrevLoc
    myDeltaAngle = GetAngle (me, myDeltaLoc)
  end if
  myDeltaLoc = the mouseLoc - myPrevLoc
  angle = GetAngle (me, myDeltaLoc)
  newAngle = myPrevAngle + angle - myDeltaAngle
  if (newAngle > 0) and (newAngle < 90) then newAngle = 360
  if (newAngle < 180) and (newAngle > 90) then newAngle = 180
  if newAngle > 360 then newAngle = 360

  mySprite.rotation = newAngle
end Turn
```

```
on Drag me -- sent by DoAction
  if myAction <> #move then
    myAction = #move
  end if

  mouseDelta = the mouseLoc - myClickLoc
  mySprite.loc = myClickLoc + mouseDelta - myDeltaLoc
end Drag
```

```
on GetLength me, vector -- sent by Grow
  deltaH = vector[1]
  deltaV = vector[2]
  return sqrt ((deltaH * deltaH) + (deltaV * deltaV))
end GetLength
```

```
on GetAngle me, slope -- sent by Turn
  deltaH = slope[1]
  deltaV = slope[2]
  if deltaH then
    slope = float (deltaV) / deltaH
    angle = atan (slope)
    if deltaH < 0 then
      angle = angle + pi
    end if
  else if deltaV > 0 then
    angle = pi / 2
  else if deltaV < 0 then
    angle = (3 * pi) / 2
  else
    angle = 0
  end if
end GetAngle
```

```
end if
-- Convert to degrees for .rotation
angle = (angle * 180) / pi

return angle
end GetAngle

-- PUBLIC METHODS (responses to #sendSprite, #sendAllSprites,
#call) --

on RotateScale_SetMessage me, theAction, theKey
-- Sets the action keys in response to an external call.
Three types of
-- syntax are supported:
-- 1) sendAllSprites #RotateScale_SetMessage, #scale, "s"
-- 2) sendAllSprites #RotateScale_SetMessage, [#action:
#scale, #key: "s"]
-- 3) sendAllSprites #RotateScale_SetMessage, \
-- [[#action: #scale, #key: "s"], [#action: #rotate, #key:
"r"]]

-- Syntax check
if listP (theAction) then
if ilk (theAction) = #propList then
if theAction.findPos (#action) then
if theAction.findPos (#key) then
theKey = theAction.key
theAction = theAction.action
else
-- Error check
return #missingParameter
end if
else
-- Error check
return #missingParameter
end if
else
-- Use recursion
repeat with thePropList in theAction
if listP (thePropList) then
RotateScale_SetMessage me, thePropList
else
return #invalidList
end if
end repeat
exit
end if
end if
-- End of syntax check

case ilk (theKey) of
#integer, #string: -- valid parameter
otherwise
-- Error check
return #invalidParameter
end case
```

```
case theAction of
#rotate: myRotateKey = RationalizeKeys (me, theKey)
#scale: myScaleKey = RationalizeKeys (me, theKey)
otherwise
-- Error check
return #invalidParameter
end case
end RotateScale_SetMessage

on RotateScale_GetReference me
-- Returns a reference to the behavior for Lingo calls
return me
end RotateScale_GetReference

-- ERROR CHECKING --

on isOKToAttach (me, aSpriteType, aSpriteNum)
case aSpriteType of
#graphic:
return getpos([#bitmap, #flash, #vectorShape],
sprite(aSpriteNum).member.type) <> 0
#script:
return FALSE
end case
end isOKToAttach

-- AUTHOR-DEFINED PARAMETERS --

on getPropertyDescriptionList me

if not the currentSpriteNum then exit

return \
[ \
#myRotateKey: \
[ \
#comment: "Press which key to rotate sprite?", \
#format: #string, \
#default: "Shift" \
], \
#myScaleKey: \
[ \
#comment: "Press which key to scale sprite?", \
#format: #string, \
#default: "Space" \
] \
] \
]
end getPropertyDescriptionList
```

1.1.1 Tích hợp dữ liệu

Quá trình tích hợp dữ liệu xây dựng sản phẩm bao gồm các bước sau:

1. Tạo các hình ảnh trong kịch bản yêu cầu bằng PaintShop Pro, PhotoShop và lưu vào hệ quản trị cơ sở dữ liệu
2. Tạo các đoạn video bằng 3D Max và hiệu chỉnh lại bằng Adobe Premiere
3. Thu âm thanh và số hoá âm thanh bằng phần mềm SnapIt
4. Đưa các dữ liệu và hình ảnh cần thể hiện, các liên kết theo đúng trình tự vào từng trang màn hình
5. Thay đổi thuộc tính (màu sắc, kích thước, độ tương phản) của các đối tượng dữ liệu trong mỗi màn hình cho phù hợp với kịch bản
6. Thông qua các hình ảnh biểu tượng đặt các tương tác thích hợp.
7. Xuất ra thành file chạy độc lập, đóng gói chương trình

1.1.2 Chương trình

Sản phẩm đa phương tiện “Hiện tượng khúc xạ ánh sáng” đã thực hiện được:

- Chương trình sử dụng 5 đoạn phim ngắn, các đoạn phim này đều đạt chất lượng cao có thể làm dữ liệu sử dụng trong các phần mềm khác. Mỗi đoạn phim minh hoạ có kích thước khung hình là 800x600, tần số âm thanh tương ứng là 11500Hz.
- Dữ liệu ảnh gồm 50 hình ảnh tĩnh
- Chương trình hỗ trợ tương tác người dùng- máy tính. Các đoạn mã nguồn có thể được sử dụng trong các sản phẩm đa phương tiện khác.
- Chương trình gồm 4 pha chính, các pha được liên kết với nhau theo đúng tiến trình sự phạm của một bài thí nghiệm vật lý ở bậc phổ thông.

1.1.3 Kiểm tra

a. Kiểm tra chi tiết từng màn hình trong sản phẩm

Sau khi xây dựng chúng tôi tiến hành kiểm tra sản phẩm cả mặt nghệ thuật và nội dung. Quá trình kiểm thử thông qua:

- Người thiết kế kiểm tra
- Người dùng kiểm thử

b. Kiểm tra toàn bộ phần mềm

- Dung lượng: dung lượng sản phẩm 30 MB
- Hỗ trợ chạy trên Internet: người dùng có thể truy cập phần mềm thông qua Internet với tốc độ 56KBps
- Chạy trên máy tính thường, yêu cầu cấu hình:
 - Bộ xử lý Pentium 75MHz
 - 8 MB RAM

- 540 MB đĩa cứng
- CD-ROM
- Bàn phím, chuột
- Yêu cầu phần mềm: vì đây là sản phẩm đóng gói, file chạy dưới dạng .exe nên không yêu cầu người dùng cài đặt bất kỳ phần mềm hay phần cứng hỗ trợ nào. Không yêu cầu nhiều tài nguyên, không xung đột với bất kỳ phần mềm nào trong máy tính.

1.1.4 In lên CD-ROM

a. Cấu trúc

Phần mềm được tổ chức thành các file và thư mục như sau:

- Tên sản phẩm: “HIỆN TƯỢNG KHÚC XẠ ÁNH SÁNG”
- File chạy: “HienTuongKhucXaAnhSang.exe”
- Thư mục:

IMAGES: lưu toàn bộ file hình ảnh

VIDEOS: lưu toàn bộ các đoạn video

AUDIOS: lưu toàn bộ các file âm thanh

b. Sao chép và tạo logo đĩa

Phần mềm “Hiện tượng khúc xạ ánh sáng” được ghi trên đĩa CD. Logo của đĩa phần mềm có biểu tượng Viện Công nghệ thông tin-Đại học Quốc gia Hà Nội và hình ảnh hiện tượng khúc xạ ánh sáng- chiếc đĩa cắm vào cốc thủy tinh có nước.

1.2. Kết quả

Phần mềm mô phỏng thí nghiệm ảo đa phương tiện “Hiện tượng khúc xạ ánh sáng” có các chỉ tiêu và yêu cầu kỹ thuật đạt được như sau:

a. Chức năng:

- Các chức năng hoạt động đơn giản, dễ sử dụng
- Bảo đảm tính sư phạm hợp lý
- Các hiện tượng mô phỏng thí nghiệm gần hiện thực và cuộc sống thường ngày
- Có tính tương tác cao, giáo viên và học sinh thực hành có thể thao tác trực tiếp trên các thí nghiệm và quan sát hiện tượng xảy ra. Người dùng có thể thực hành theo video. Trong mỗi thí nghiệm, người dùng sẽ được đọc và xem hình minh họa hiện tượng bằng video. Người dùng có thể thao tác lại các thí nghiệm đó theo ý muốn và xem hiện tượng xảy ra, sau đó có thể tự đưa ra

nhận xét và ghi nhớ vào trong từng thí nghiệm hoặc có thể xem kết luận trong từng hiện tượng vật lý.

- Ngôn ngữ thể hiện dễ hiểu với học sinh theo chuẩn sách giáo khoa giảng dạy.
- Đảm bảo nội dung cần chuyển tải trong bài thí nghiệm. Tất cả các thí nghiệm và hiện tượng đều có nội dung chính gồm:
 - Tên thí nghiệm hoặc hiện tượng
 - Hình ảnh minh họa phân tương tác cho phép người dùng tự tạo hiện tượng
 - Tạo liên kết tới các hiện tượng, thí nghiệm liên quan khác (nếu có). Nút bấm quay về màn hình trước, sau, màn hình chủ.
 - Âm thanh và khung văn bản hiện lời nhận xét, kết luận đối với mỗi hiện tượng sẽ hiện thị khi người dùng kích chuột vào các biểu tượng âm thanh và văn bản.

b. Giao diện

- Thân thiện, dễ sử dụng
- Tương tác thông qua menu
- Làm nổi bật tính trực quan, tính đa dạng trong thể hiện nội dung
- Các hình ảnh minh họa nút bấm đã được xử lý cho phép người dùng có thể dễ dàng nhận biết chức năng của chúng để thực hiện.

c. An toàn và bảo mật

- Sản phẩm được đóng gói dưới dạng file chạy .exe, đảm bảo tính bảo mật cao
- Có cơ chế bảo vệ cơ sở dữ liệu đã xây dựng

d. Tính chuẩn hoá

- Mọi dữ liệu đều được chuẩn hóa theo chuẩn quốc tế và chuẩn công nghiệp, phù hợp với phần mềm sử dụng, góp phần tăng năng suất và hiệu quả xây dựng sản phẩm.
- Chuẩn hoá trong thiết kế giao diện với người dùng.
- Thống nhất sử dụng ngôn ngữ Lingo là ngôn ngữ chính trong việc lập trình tạo sản phẩm

e. Tính mở, tính khả chuyển

- Phần mềm “Hiện tượng khúc xạ ánh sáng” được chia thành các module nhỏ, có thể sử dụng lại các module này trong quá trình xây dựng các sản phẩm khác.

- Bảo đảm tính khái quát cho từng lớp các bài thí nghiệm tương tự
- Người thiết kế xây dựng sau có thể dựa vào quy trình xây dựng phần mềm mô phỏng thí nghiệm ảo đa phương tiện tổng quát chung để tạo sản phẩm mới.

f. Môi trường và công cụ cài đặt

- Chương trình được cài đặt trên hầu hết các hệ điều hành phổ dụng Windows NT, Windows 9X và 2000 mà không cần yêu cầu tài nguyên đặc biệt nào.

g. Tài liệu kỹ thuật đi kèm

- Tài liệu hướng dẫn sử dụng phần mềm “Hiện tượng khúc xạ ánh sáng”