**LỜI MỞ ĐẦU**

Thực tế ảo là một thuật ngữ mới xuất hiện khoảng đầu thập kỷ 90, nhưng ở Mỹ và châu Âu thực tế ảo (Virtual Reality) đã và đang trở thành một công nghệ mũi nhọn nhờ khả năng ứng dụng rộng rãi trong mọi lĩnh vực (nghiên cứu và công nghiệp, giáo dục và đào tạo, thương mại và giải trí,..) và tiềm năng kinh tế, cũng như tính lưỡng dụng (trong dân dụng và quân sự) của nó. Tại Việt Nam, tuy là một lĩnh vực mới nhưng đã có những công trình rất hữu ích như: tái hiện lại con Sao La hay một Văn Miếu Quốc Tử Giám ảo mà ta có thể đi lại quan sát trong đó. Chính vì tầm quan trọng cũng như khả năng ứng dụng to lớn đó nên việc nghiên cứu về thực tại ảo là vô cùng cần thiết. Và trên cơ sở đó có thể xây dựng một ứng dụng thực tại ảo hoàn chỉnh.

Chính vì vậy nhóm chugns em đã chọn đề tài “Xây dựng ký túc xá ĐH CNHN bằng VRML”.

Nội dung bao gồm 3 chương :

* Chương 1: Tổng quan về thực tại ảo.
* Chương 2: Ngôn ngữ VRML.
* Cương 3: ứng dụng VRML trong việc xây dựng ký túc xá trường ĐH CNHN.

MỤC LỤC

[Chương 1. Tổng quan về thực tại ảo 3](#_Toc436937017)

[1.1. Thực tại ảo là gì 3](#_Toc436937018)

[1.2. Lịch sử phát triển của công nghệ thực tại ảo 4](#_Toc436937019)

[1.3. Các đặc tính chính của VR 4](#_Toc436937020)

[1.4. Các thành phần của một hệ thống VR 5](#_Toc436937021)

[1.4.1. Phần cứng(Hardware) 5](#_Toc436937022)

[1.4.2. Phần mềm(Software) 5](#_Toc436937023)

[1.5. Các thiết bị cơ bản 6](#_Toc436937024)

[1.5.1. Thiết bị định hướng và chuyển động 6](#_Toc436937025)

[1.5.2. Thiết bị tương tác và phản hồi 8](#_Toc436937026)

[1.6. Một số ứng dụng chính của VR 9](#_Toc436937027)

[1.6.1. Quân sự 9](#_Toc436937028)

[1.6.2. Giáo dục 10](#_Toc436937029)

[1.6.3. Xây dựng 11](#_Toc436937030)

[1.6.4. Y học 11](#_Toc436937031)

[Chương 2. Ngôn ngữ VRML 12](#_Toc436937032)

[2.1. Giới thiệu về VRML 12](#_Toc436937033)

[2.1.1. VRML là gì 12](#_Toc436937034)

[2.1.2. Định nghĩa về VRML 12](#_Toc436937035)

[2.1.3. Lịch sửa ra đời và phát triển của VRML 12](#_Toc436937036)

[2.1.4. Đặc điểm cơ bản của VRML 12](#_Toc436937037)

[2.2. Các vấn đề cơ bản của VRML 12](#_Toc436937038)

[2.2.1. Các thành phần cơ bản của VRML 12](#_Toc436937039)

[2.2.2. Công cụ hiển thị VRML 12](#_Toc436937040)

[2.2.3. Tập tin của VRML 12](#_Toc436937041)

[2.3. Tìm hiểu chi tiết về VRMl 12](#_Toc436937042)

[2.3.1. Xây dựng đối tượng hình học cơ bản 12](#_Toc436937043)

[2.3.2. Xây dựng một số hình học phức tạp 12](#_Toc436937044)

[2.3.3. Các phép biến đổi trong VRML 12](#_Toc436937045)

[2.3.4. Màu sắc trong VRML 12](#_Toc436937046)

[2.3.5. Nhóm Node 12](#_Toc436937047)

[2.3.6. Một số phương pháp vẽ trong VRML 12](#_Toc436937048)

[2.3.7. Texture Mapping 12](#_Toc436937049)

[2.3.8. Script 12](#_Toc436937050)

[Chương 3. Ứng Dụng VRML trong việc xây dựng ký túc xá ĐH CNHN 12](#_Toc436937051)

[3.1. Bài toán 12](#_Toc436937052)

[3.2. Yêu cầu đặt ra và hướng giải quyết 12](#_Toc436937053)

[3.2.1. Xây dựng ký túc xá 12](#_Toc436937054)

[3.2.2. Trang trí ký túc xá 12](#_Toc436937055)

[3.2.3. Xây dựng các đối tượng tỏng ký túc xá 12](#_Toc436937056)

[3.2.4. Di chuyển bên trong ký túc xá 12](#_Toc436937057)

[3.2.5. Chọn đối tượng 12](#_Toc436937058)

[3.2.6. Ánh sáng 12](#_Toc436937059)

[3.2.7. Đóng mở cửa 12](#_Toc436937060)

[3.2.8. Tối ưu khung hình 12](#_Toc436937061)

[3.3. Kết quả đạt được và hướng phát triển 12](#_Toc436937062)

[3.3.1. Kết quả 12](#_Toc436937063)

# Tổng quan về thực tại ảo

## Thực tại ảo là gì

Thực tế ảo-Virtual Reality( VR) là một hệ thống mô phỏng trong đó đồ họa máy tính được sử dụng để tạo ra một thế giới "như thật". Hơn nữa, thế giới "nhân tạo" này không tĩnh tại, mà lại phản ứng, thay đổi theo ý muốn (tín hiệu vào) của người sử dụng (nhờ hành động, lời nói,..). Điều này xác định một đặc tính chính của VR, đó là tương tác thời gian thực (real-time interactivity). Thời gian thực ở đây có nghĩa là máy tính có khả năng nhận biết được tín hiệu vào của người sử dụng và thay đổi ngay lập tức thế giới ảo. Người sử dụng nhìn thấy sự vật thay đổi trên màn hình ngay theo ý muốn của họ và bị thu hút bởi sự mô phỏng này.

Điều này chúng ta có thể nhận thấy ngay khi quan sát trẻ nhỏ chơi video game. Theo báo Bild (Đức), có hai trẻ nhỏ ở Anh bị thu hút và mải mê chơi Nintendo đến nỗi ngay cả khi nhà chúng đang bị cháy cũng không hề hay biết! Tương tác và khả năng thu hút của VR góp phần lớn vào cảm giác đắm chìm (immersion), cảm giác trở thành một phần của hành động trên màn hình mà người sử dụng đang trải nghiệm. Nhưng VR còn đẩy cảm giác này "thật" hơn nữa nhờ tác động lên tất cả các kênh cảm giác của con người. Trong thực tế, người dùng không những nhìn thấy đối tượng đồ họa 3D nổi (như hình nổi ở trang cuối báo Hoa học trò đã đăng trước kia), điều khiển (xoay, di chuyển,..) được đối tượng trên màn hình (như trong game), mà còn sờ và cảm thấy chúng như có thật. Ngoài khả năng nhìn (thị giác), nghe (thính giác), sờ (xúc giác), các nhà nghiên cứu cũng đã nghiên cứu để tạo các cảm giác khác như ngửi (khứu giác), nếm (vị giác). Tuy nhiên hiện nay trong VR các cảm giác này cũng ít được sử dụng đến.

Từ các phân tích trên, chúng ta có thể thấy định nghĩa sau đây của c. Burdea và p. Coiffet về VR là tương đối chính xác: VR- Thực Tế Ảo là một hệ thống giao diện cấp cao giữa Người sử dụng và Máy tính. Hệ thống này mô phỏng các sự vật và hiện tượng theo thời gian thực và tương tác với người sử dụng qua tổng hợp các kênh cảm giác. Đó là ngũ giác gồm: thị giác, thính giác, xúc giác, khứu giác, vị giác.

## Lịch sử phát triển của công nghệ thực tại ảo

Khái niệm thực tế ảo đã có trong nhiều thập niên nhung nó chỉ thực sự đuợc nhận thức vào đầu những năm 90. Vào giữa những năm 50 Morton Heilig (Mỹ) đã phát minh ra thiết bị mô phỏng SENSORAMA. Đó là 1 thiết bị điều khiển 1 nguời sử dụng gồm có : một màn hình thực thể kính, quạt, máy tạo mùi, loa âm thanh và 1 chiếc ghế có thể di chuyển đuợc. Ông cũng phát minh ra màn hình truyền hình đuợc gắn vào đầu để có thể xem phim 3D. Tuy là những sản phẩm phục vụ cho điện ảnh nhung những khái niệm của Heilig đã trở thành tiền đề cho VR sau này.

Những kỹ su của Công ty Philco là những nguời đầu tiên phát triển HMD vào 1961, gọi là Headsight. Cái mũ sắt bao gồm một màn ảnh và hệ thống theo dõi video đã những kỹ su liên kết tới một hệ thống camêra mạch đóng. Họ dụ định sử dụng HMD trong các tình huống nguy hiểm - một nguời có thể quan sát một môi truờng thực sự từ xa, điều chỉnh góc quay camera bằng cách quay đầu. Bell Laboratories đã sử dụng HMD cho những phi công lai máy bay trục thăng. Họ liên kết HMD với những camera hồng ngoại gắn bên ngoài máy bay giúp phi công có thể nhìn rõ ngay cả trong môi truờng thiếu ánh sáng.

Vào 1965, một nhà khoa học máy tính có tên Ivan Sutherland hình dung điều mà ông ta gọi là "Ultimate Display". Sử dụng hiển thị này, một nguời có thể thấy một thế giới ảo hiện ra nhu thế giới vật lý thật. Điều này đã định huớng toàn bộ tầm nhìn về VR. Khái niệm của Suntherland bao gồm :

❖ Một thế giới ảo mà ta có thể quan sát thông qua một HMD

❖ Một máy tính để duy trì các mô hình trong thời gian thục

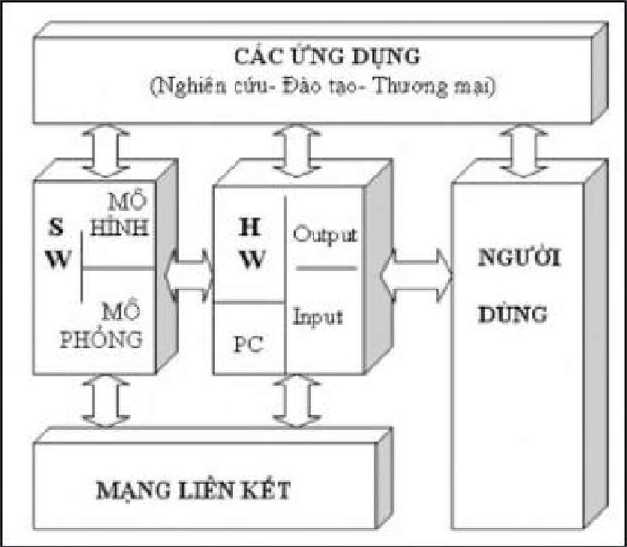
❖ Các khả năng cho nguời sử dụng để thao tác những đối tuợng thục tế một

cách trực quan nhất.

## Các đặc tính chính của VR

Như trên đã trình bày, 2 đặc tính chính của VR là Tương tác và Đắm chìm, đây là hai 'T' (Interactive, Immersion) mà nhiều người đã biết. Tuy nhiên VR cần có 1 đặc tính thứ 3 mà ít người để ý tới. VR không chỉ là một hệ thống tương tác Người- Máy tính, mà các ứng dụng của nó còn liên quan tới việc giải quyết các vấn đề thật trong kỹ thuật, y học, quân sự,...Các ứng dụng này do các nhà phát triển VR thiết kế, điều này phụ thuộc rất nhiều vào khả năng Tưởng tượng của con người, đó chỉnh là đặc tính T’ (Lnagination) thứ 3 của VR. Do đó có thể coi VR là tổng hợp của 3 yếu tố: Tương tác- Đắm chìm- Tưởng tượng, (3 I trong tiếng Anh: Interactive- Immersion- Imagination)

## Các thành phần của một hệ thống VR

Tổng quát một VR bao gồm những thành phần sau

*Hình 1.1 Các thành phần của một hệ thông VR*

### Phần cứng(Hardware)

Phần cứng của một VR bao gồm:

❖ Máy tính (PC hay Workstation với cấu hình đồ họa mạnh).

❖ Các thiết bị đầu vào (Input devices): Bộ dò vị trí (position tracking) để xác định vị trí quan sát. Bộ giao diện định vị (Navigation interíaces) để di chuyển vị trí người sử dụng. Bộ giao diện cử chỉ (Gesture interíaces) như găng tay dữ liệu (data glove) để người sử dụng có thể điều khiển đối tượng.

❖ Các thiết bị đầu ra (Output devices): gồm hiển thị đồ họa (như màn hình, HDM,..) để nhìn được đối tượng 3D nổi. Thiết bị âm thanh (loa) để nghe được âm thanh vòm (như Hi-Fi, Surround,..). Bộ phản hồi cảm giác (Haptic feedback như găng tay,..) để tạo xúc giác khi sờ, nắm đối tượng. Bộ phản hồi xung lực (Force Feedback) để tạo lực tác động như khi đạp xe, đi đường xóc,...

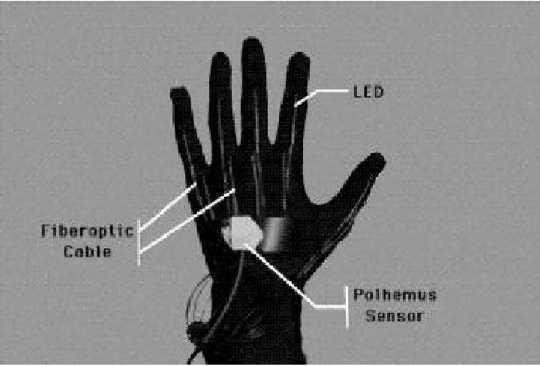
### Phần mềm(Software)

Phần mềm luôn là linh hồn của VR cũng như đối với bất cứ một hệ thống máy tính hiện đại nào. về mặt nguyên tắc có thể dùng bất cứ ngôn ngữ lập trình hay phần mềm đồ họa nào để mô hình hóa (modelling) và mô phỏng (simulation) các đối tượng của VR. Ví dụ như các ngôn ngữ (có thể tìm miễn phí) OpenGL, C++, Java3D, VRML, X3D,. ..hay các phàn mềm thương mại như WorldToolKit, PeopleShop,... Phần mềm của bất kỳ VR nào cũng phải bảo đảm 2 công dụng chính: Tạo hình vào Mô phỏng. Các đối tượng của VR được mô hình hóa nhờ chính phần mềm này hay chuyển sang từ các mô hình 3D (thiết kế nhờ các phần mềm CAD khác như AutoCAD, 3D Studio,..). Sau đó phần mềm VR phải có khả năng mô phỏng động học, động lực học, và mô phỏng ứng xử của đối tượng.

## Các thiết bị cơ bản

### Thiết bị định hướng và chuyển động

* **DataGloves**

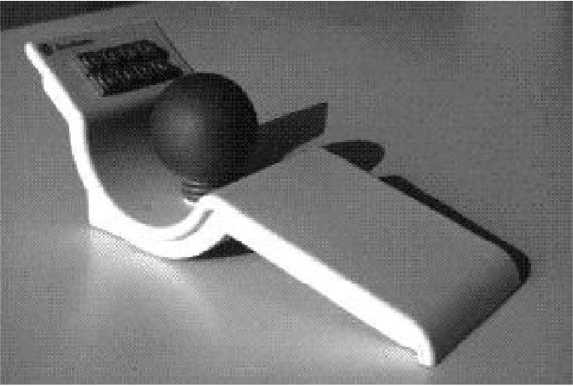
Thiết bị đo lường bàn tay phải cảm nhận được cả độ cong của các ngón tay và vị trí, sự định hướng của cổ tay trong thời gian thực. Thiết bị thương mại đầu tiên là DataGloves từ viện nghiên cứu VPL. DataGloves bao gồm 1 găng tay nylon nhẹ có các cảm bỉến quang học được gắn ở các ngón tay.

*Hình 1.2 DataGloves 3D Mouse and SpaceBall*

* **3D Mouse va SpaceBall**

*Hình 1.3 3D Mouse và SpaceBall*

Chuột Logitech 3D dựa ừên một mảng các vị trí siêu âm tham chiếu, đó là 1 cái kiềng gồm 3 loa siêu âm đặt ở 3 góc tam giác phát ra tín hiệu siêu thanh. Nó được sử dụng để theo dõi thiết bị thu, định hướng và chuyển động. Nó qui định thành phần của tỷ lệ gửi ra trong tất cả 6. mức tự do: X, Y, z, Pitch, Yaw, và Roll.

*Hình 1.4 Mouse*

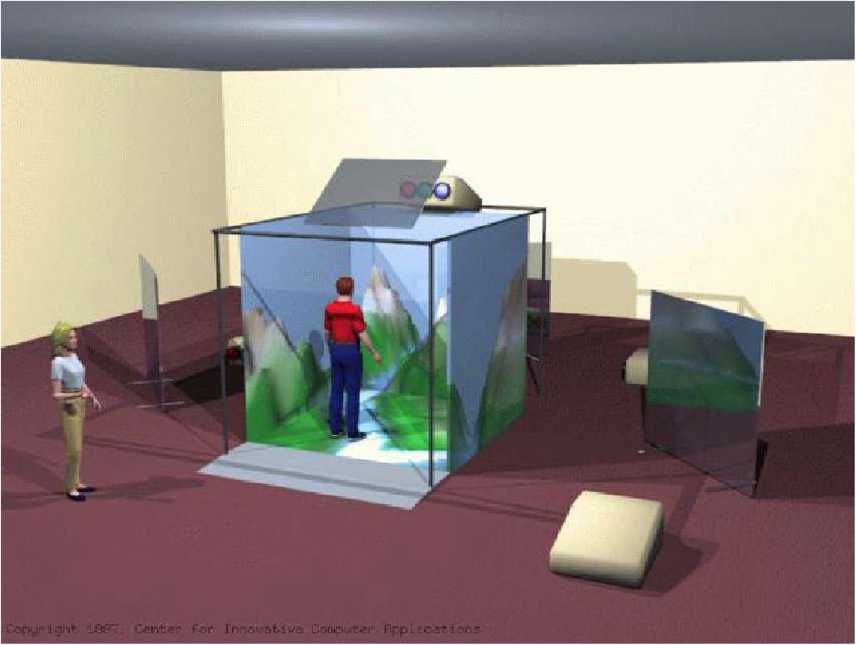
* Shutter glasses

*Hình 1.5 Shutter glasses*

* **Head-Mounted Displays**

*Hình 1.6 Head-Mounted Displays*

* **Cave**

CAVE là 1 nhà hát có kích thước 10 X 10 X 9 được đặt bên ừong 1 phòng lớn hơn có kích thước 35 X 25 X 13. Phòng bên ngoài phải được chiếu sáng trong suốt quá trình sử dụng CAVE. Các bức tường của CAVE được tạo bởi các màn chiếu và sàn nhà cũng là một màn chiếu thẳng đứng. Máy chiếu độ phân giải cao hiển thị hình ảnh lên toàn bộ nhứng màn ảnh khác bằng các tấm gương phản chiếu. Nguwoif dùng sẽ đi vào bên trong CAVE và đeo 1 chiêc kinh đặc biệt để có thể nhìn thấy những hình ảnh 3 chiều mà CAVE hiển thị. Với những chiếc kính này người dùng có thể thấy các đối tượng thực sự nổi trong không khí và có thể đi lại xung quanh chúng. Điều này la hoàn toàn khả dĩ với các cảm biến điện tử. Khung của CAVE được làm từ i-nox không từ tính để có thể can thiệp một cách tốt nhất vào các cảm biến điện từ. Khi một người đi lại trong CAVE, chuyển động của họ được theo dối bởi các cảm biến này và vỉdeo sẽ điều chỉnh cho phù hợp. Máy tính sẽ kiểm soát việc này của CAVE cũng như cả khía cạnh âm thanh nữa. Có rất nhiều loa được đặt trong CAVE dưới nhiều góc độ giúp cho không chỉ có hình ảnh 3 chiều mà có cả âm thanh 3 chiều nữa.

*Hình 1.7 Cave*

### Thiết bị tương tác và phản hồi

Các thiết bị này cảm nhạn một số nhân tổ Sâu của thiết bị khác gây ra: nhiệt độ, vận tốc dỉ chuyển, sự chuyển động, áp lực và các ngoại lực khác.

* **CyberTouch**



*Hình 1.8 CyberTouch*

* **CyberGrasp**

*Hình 1.9 CyberGrasp*

## Một số ứng dụng chính của VR

Tại các nước phát triển, chúng ta có thể nhận thấy VR được ứng dụng trong mọi lĩnh vực: Khoa học kỹ thuật, kiến trúc, quân sự, giải trí,... và đáp ứng mọi nhu cầu: Nghiên cứu- Giáo dục- Thương mại. Y học là lĩnh vực ứng dụng truyền thống của VR. Bên cạnh đó VR cũng được ứng dụng ừong giáo dục, nghệ thuật, giải trí. Trong lĩnh vực quân sự, VR cũng được ứng dụng rất nhiều ở các nước phát triển. Bên cạnh các ứng dụng truyền thống ở trên, cũng có một số ứng dụng mới nổi lên ừong thời gian gần đây của VR như: VR ứng dụng trong sản xuất, VR ứng dụng trong ngành rôbốt, VR ứng dụng ừong hiển thị thông tin (thăm dò dầu mỏ, hiển thị thông tin khối,....) VR có tiềm năng ứng dụng vô cùng lớn. Cổ thể nói tóm lại một điều: Mọi lĩnh vực "có thật " trong cuộc sống đều có thể ứng dụng "thực tế ảo" để nghiên cứu và phát triển hoàn thiện hơn.

### Quân sự

Với việc phát triển của VR, các binh sĩ sẽ được huấn luyện 1 cách trực quan nhất các lã năng cân thiết như : lái máy bay, lai xe tăng, . .. trước khỉ tham gia công việc thực tế. Điều này vừa bảo đảm an toàn cho binh sĩ, vừa tiết kiệm được chi phí cho các khoa huấn luyện thực tế. Lầu Năm Góc vừa đưa ra quyết định sẽ đàu tư 36 triệu USD cho quân đội Mỹ để phát triển một game đặc biệt nhằm huấn luyện binh sĩ chống lại khủng bố dưới dạng chiến thuật thực tế ảo.Với hệ thống trò choi đặc biệt này, những binh sĩ có thể tập luyện những bài tập của mình ngay tại nhà nhằm chống lại những tình huống có thể phát sinh ra ừong thực tế. Đây sẽ là một game rất sống động, có tình hành động cao với môi trường và bối cảnh bám sát với thực tế. Những người lính sẽ phải vận dụng tất cả những kỹ năng đã được rèn giũa trong quân đội.

*Hình 1.10 Bỉnh lính học nhảy dù băng thực tê ảo*

### Giáo dục

Ở các nước phương Tây việc ở nhà học qua Internet không còn là điều mới mẻ nữa. Và công nghệ VR sẽ làm cho việc này ừở nên thú vị hơn rất nhiều. Giống như một game MMORPG bạn điều khiển 1 nhân vật đại diện cho bạn đi lại ừong 1 trường học ảo được xây dựng trên máy tính. Bạn có thể tham gia vào bất cứ lớp học ảo nào mà bạn thích, nối chuyện với nhưng thành viên khác trong lớp.



*Hình 1.11 Cảnh ưong một lớp học ảo*

### Xây dựng

Bạn muốn xây nhà. Bạn thuê một kiến trúc sư thiết kế cho ngôi nhà tương lai của bạn. Anh ta hoàn thành nó trên bản vẽ và liệu bạn có thể tưởng tượng ra nó thế nào không ?CÓ thể nhưng chắc là không thể chính xác được. Và khi hoàn thành thì chưa chắc nó đã đủng ý của bạn. Giờ đây ngôi nhà đó được xây dựng ưên máy tình, bạn cố thể đi lại khắp nơi trong nhà, xem xét tưng ngõ ngách nhỏ nhất.

### Y học

Thực tại ảo giải quyết được rất nhiều vấn đề ưong y học: cung cấp môi trường thực hành cho nghiên cứu và học tập, rất hữu ích trong việc mô phỏng các ca phẫu thuật ừánh gây rủi ro ừong thực tế

Như vậy thực tại ảo có ứng dụng trong hầu hết các lĩnh vực của cuộc sống. Qua đó cũng nhận thấy được ý nghĩa to lớn của việc ứng dụng thực tại ảo, bởi những vấn đề khó khăn mà nếu không có thực tại ảo thì rất khó giải quyết hoặc hiểu quả không cao mà chi phỉ tốn kém.

# Ngôn ngữ VRML

## Giới thiệu về VRML

### VRML là gì

VRML (Virtual Reality Modeling Language) là ngôn ngữ mô hình hóa thực tế ảo, một định dạng tập tin được sử dụng trong việc mô tả các thế giới và các đối tượng đồ họa tương tác ba chiều. VRML được thiết kế dùng trong môi trường Internet, Intranet và các hệ thống máy khách cục bộ (local Client). VRML còn được dự trù trở thành một chuẩn trao dổi đa năng cho đồ họa ba chiều tích họp và truyền thông đa phương tiện. VRML có thể được sử dụng trong rất nhiều lĩnh vực ứng dụng chẳng hạn như trực quan hóa các khái niệm khoa học và kỹ thuật, trình diễn đa phương tiện, giải trí và giáo dục, hỗ trợ web và chia sẻ các thế giới ảo.

Ra đời phiên bản đàu tiên vào tháng 10 năm 1994 (VRML 1.0), cho đến nay VRML đã phát triển tới phiên bản 2.0 với các chức năng mạnh mẽ, nhanh chóng trở thành chuẩn phát triển cho nhiều chương trình đồ họa. VRML là ngôn ngữ Intemer 3D dùng để phát triển đồ họa 3D trên Web, có cấu trúc chặt chẽ, với khả năng mạnh mẽ, giúp cho việc xây dựng các ứng dụng 3D một cách nhanh chóng và chân thực nhất.

### Định nghĩa về VRML

Ngôn ngữ VRML là ngôn ngữ sử dụng mô hình phân cấp trong việc thể hiện các tương tác với các đối tượng của mô hình, VRML được sử dụng để phát triển những hình ảnh 3D và qung cảnh trên Web. Các fĩle VRML có kích thức nhỏ, thường không quá 1Mb.

Ngôn ngữ mô hình hóa thực tại ảo VRML là một chuẩn không chính thức để mô tả thực tế ảo mà không phụ thuộc vào hệ điều hành thông qua Internet. Chỉ với một fĩle text bạn có thể mô tả, tương tác, điều khiển một thế giới ảo mà không bị hạn chế nhiều.

VRML cho phép truyền đi trong mạng những hình ảnh 3D. Với kích thước khả nhỏ so với băng thông, phần lớn giới hạn trong khoảng 100 - 200Kb nên các file VRML được tmyền đi một cách khá dễ dàng. Nếu HTML là định dạng văn bản thì VRML là định dạng đối tượng 3D. Hiện nay VRML có lợi thế là sự đon giản, hỗ trợ dịch vụ web3D.

### Lịch sửa ra đời và phát triển của VRML

VRML đã trở thành một ngôn ngữ chuẩn cho việc mô phỏng tương tác thế giới 3D tren Web. Với mục đích xây dựng định dạng chuẩn cho phép mô tả thế giới thực trên máy tính mà cho phép chạy trên môi trường web, VRML đã trở thảnh chuẩn ISO từ năm 1997.

VRML ra đời vào mùa xuân năm 1994 ở hội nghị www được tổ chức đầu tiên tại Gieneva, Thụy Sĩ. Tim Bemers-Lee và Dave Raggett đã tổ chức ra phiên họp có tên là Birds of a Feather (BOF) để mô tả giao diện thực tế ảo trên www. Nhiều thành viên tham dự phiên họp BOF đã mô tả nhiều dự án thực hiện việc xây dựng các công cụ hiển thị đồ họa 3D cho phép có nhiều thao tác hữu ích trên Web. Những thành viên này đã nhất trí đồng ý sự cần thiết cho các công cụ này có một ngôn ngữ chung, phổ biến cho đạnh dạng, xác định việc mô tả thế giới 3D và các siêu liên kết www. Vì thế cụm từ “the Virtual Reality Markup Language” ra đời, từ Markup sau đó đã được đổi thành Modelling để phản ánh bản chất tự nhiên của VRML.

Sau phiên họp BOF một thời gian ngắn thì tổ chức www-vrm1 mailing list được thành lập để tập trung vào xây dựng phiên bản VRML đầu tien. Sự hưởng ứng lời mời của tổ chức này kéo dài đến một tuần và có trên một nghìn khách mời tham dự. Tại buổi họp chủ tịch Mark Pesce đã thông báo ý kiến của mình là đưa ra phiên bản phác thảo xây dựng VRML đã có sẵn ở hội nghị mùa xuân năm 1994 được tổ chức mới cách đó 5 tháng. Bản phác thảo của Mark Pesce đã có được sự đồng ý chung.

Vào tháng 3/ 1995 thì có một thảo luận trên www-vrml mailing list liên quan đến việc tạo ra những tương tác của người sử dụng với hoạt cảnh và tất cả mọi người đều đi đến thống nhất ý kiến những thứ mới đưa ra đó thực sự cần thiết cho VRML. Công ty Silicon Graphics cộng tác với hãng Sony Research và Mitra để đưa ra phiên bản mới cho VRML. Bản đệ trình của Silicon Graphics có tên là Moving Worlds đến tổ chức Request for Proposals cho việc xây dựng phiên bản mới VRML, bản đệ trình này là một minh chứng cho sự cộng tác thành công của tất cả các thành viên của Silicon Graphics, Sony và Mitra. Năm 1996 tại New Orleans, phiên bản đầu tiên của VRML 2.0 được đưa ra.

Vào tháng 7/1996 tổ chức tiều chuẩn quốc tế ỢSO) đã thông nhất ý kiến phiên bản năm 1996 của VRML 2.0 như là một đề xuất mà sẽ được đưa ra xem xet vào thang 4/1997. Sau khi bỏ phiếu về chuẩn ISO thì VRML97 được đưa ra như một chuẩn ISO vào năm 1997.

### Đặc điểm cơ bản của VRML

Tiếu chuẩn cho việc xác định đối tượng 3D, quang cảnh và cho sự liên kết các mô hình với nhau là:

❖ Không phụ thuộc phần cứng: có thể chạy hên các máy tính do các nhà sản xuất khác nhau chế tạo.

❖ Có thể mở rộng: có thể chấp nhận các lệnh mới do người sử dụng thêm vào hoặc quy định.

❖ Thao tác được thế giới ảo thông qua môi trường Internet có băng thông thấp.

Cùng với VRML thì có thể xem thông tin trong mô hình 3D hên Internet. VRML được thiết kế dành riêng cho việc hiển thị thế giới 3D và không phải là sự mở rộng của HTML. HTML có khả năng hiển thị các đối tượng tĩnh và động, các đối tượng multimedia cùng với các siêu liên kết khác đến các media khác như là văn bản, âm thanh, phim và hình ảnh.

## Các vấn đề cơ bản của VRML

### Các thành phần cơ bản của VRML

Gồm các trình duyệt (Browses) và bộ soạn thảo dành riêng cho ngôn ngữ VRML. Các file chỉ có thể đọc nếu hệ thống có trình duyệt VRML.

Trình duyệt VRML cũng giống như trình duyệt Intemet( Internet Explorer hay Fire Fox) và được tích hợp trong các trình duyệt này.

Bộ soạn thảo VRML cho phép người dùng gõ mã VRML. Hiện nay có nhiều bộ soạn thảo nhưng VRML Pad là khá thông dụng khi có thể cho xem trực tiếp kết quả mà không cần qua trình duyệt Internet.

### Công cụ hiển thị VRML

Để hiển thị các file VRML thì ta sử dụng chương trình Cortona VRML Client của hãng Parallrl Graphics. Chương trình sẽ giúp bạn thuận tiện hơn khi xem cắc mô hình ảo trên máy tính một cách trực quan sinh động.

Yêu cầu trước khi cài đặt Cortona VRML Client:

- Hệ điều hành Microsoíì® Windows® ME/2000/XP

- Trình duyệt Web Internet Explorer 6.0 trở lên, Netscape Navigator 8.0 ừở lên, Mozilla Firefox 1.5 trở lên, Opera 8.5 trở lên

- CPU Pentium® II300 MHz trở lên.

- RAM tối thiểu 64 MB.

- Độ phân giải màn hình tối thiểu 1024x768.

- Card đồ họa hỗ trợ 3D và cài đặt DirectX 9

Cortona VRML Client tương thích với hầu hết các trình duyệt như Internet Explorer, Netscape Browser, Mozilla, Mozilla Firefox và các công cụ văn phòng như Word, PowerPoint...

Tính năng của Cortona VRML ClientCortona VRML Client sẽ trình diễn toàn bộ mô hình 3D trên máy tính một cách hoàn hảo như khi người tạo ra đó. ứng dụng hoàn chỉnh trên toàn bộ hiệu ứng trên nhiều hệ thống như Flash, DirectX9, MPEG4... Khi bạn truy xuất vào một ứng dụng VRML, toàn bộ hình mô phỏng sẽ được trình diễn tưomg tác trên nền 3D dạng mở. Rất ấn tượng và bắt mắt.

### Tập tin của VRML

Tập tin của VRML có thể tạo ra từ nhiều cách: dùng một trình soạn thảo văn bản như Nopepad để soạn thảo, sau đó lưu file với phần mở rộng là. wrl. Hoặc ta co thể soạn thảo trên một trình soạn thảo dành riêng cho VRML như VRML Pad và chạy trực tiếp ứng dụng.

Một fĩle VRML gồm có các phần như: header, scene graph, prototype và event routing.

❖ Header: dùng để nhận dạng tập tin VRML và cách mã hóa. Header của fĩle VRML bắt đầu bằng dấu #. Ngoài lần xuất hiện đầu tiên ra thì dấu # đánh dấu những gì theo sau nó là phần chú thích. File tiêu đề của VRML có dạng: #VRML VI.0 ascii dành riêng cho phiên bản VRML 1.0 và #VRML V2.0 utf-8 dành cho phiên bản 2.0.

❖ Scene Graph: chứa những node mô tả các đối tượng và các thuộc tính đi kèm. Nó gần như một cây phả hệ gồm các nhóm đối tượng.

❖ Prototype: cho phép một tập các nút kiểu VRML được mở rộng bởi người sử dụng. Các định danh kiểu này có thể được bao hàm trong fĩle (mà chúng được sử dụng) hay định nghĩa ở bên ngoài (fĩle đó).

❖ Event routing: một số nút có thể phát sinh những sự kiện đáp trả những thay đổi môi trường do tưcmg tác phía người dùng. “Event routing” cho phép một sự kiện phát sinh được truyền đến các “đích”- những nút trong hệ thống, từ đó gây ra những thay đổi cho riêng nút đó và hệ thống.

## Tìm hiểu chi tiết về VRMl

### Xây dựng đối tượng hình học cơ bản

* Box - hình hộp

Ví dụ :

Shape{

geometry Box{ size 2.0 2.0 2.0 }

}

Các tham số

* Size X Y z (chiều rộng, chiều cao, chiều sâu)
* Cone - hình nón

Ví dụ :

Shape{

Geometry Cone{ height 2.0 bottomRadius 1.0 bottom TRUE side TRUE}

}

Các tham số

* height X : chiều cao của hình nón
* bottomRadius Y : bán kính của đáy
* bottom TRUE / FALSE : hiện / ẩn đáy
* side TRUE / FALSE : hiện / ẩn mặt bên
* Cylinder : vẽ hình trụ

Ví dụ :

Shape{

Geometry Cylinder{

height 2.0

radius 1.0

bottom TRUE

top TRUE

side TRUE}

}

Các tham số

* height X : chiều cao
* radius Y : bán kính mặt đáy
* bottom TRUE / FALSE : hiện / ẩn mặt đáy
* top TRUE / FALSE : hiện / ẩn mặt nắp
* side TRUE / FALSE : hiện / ẩn các mặt bên
* Sphere : vẽ hình cầu

Ví dụ :

Shape{

Geometry Cylinder{

Radius 1.0

}

}

Các tham số

* Radius X : bán kính

### Xây dựng một số hình học phức tạp

* Text:vănbản

Cú pháp

Shape{

Geometry Text{

String [“Text”,”Shape”]

fontStyle FontStyle{ family “SERIF” style “BOLD”}

}

}

Các tham số

* string [“Nội dung văn bản”]
* family SERIP / SANS / TYPEWRITER
* style BOLD / ITALIC / BOLDITALIC / PLAIN
* size X : chiều cao của chữ
* spacing Y :
* justify FIRST / BEGIN / MIDDLE / END
* horizontal TRUE / FALSE : trình bày ngang / dọc màn hình
* leftToRight TRUE / FALSE : trình bày từ phải / trái sang
* topToBottom TRUE / FALSE : trình bày từ trên / dưới

Ví dụ:

Shape{

Geometry Text{

String [“Hi all, Today is fine.”]

fontStyle FontStyle{

family “TYPEWRITER” style “plain”

horizontal TRUE justify “MIDDLE”

leftToRight TRUE size 0.9

spacing 1.0 topToBottom TRUE

}

Lenght [6] maxExtent 6

}

}

* Xây dựng các đường thẳng trong hệ tọa độ ba chiều

Thẻ IndexedLineSet xác định một tập họp các đường thẳng trong hệ tọa độ không gian ba chiều của thế giới VRML và tập họp các màu tương ứng cho các đường thẳng đó.

Thẻ này có các thuộc tính:

* Coord và color: số thành phần ữong trường color không nhất thiết phải bằng với số thành phần trong trường coord. Chẳng hạn khi colorPerVexter nhận giá trị là FALSE.
* Coordlndex: bao gồm một dãy chỉ số thứ tự cho các điểm ảnh tạo nên đường thẳng. Ví dụ colorlndex [0 12 3] có nghĩa là đường thẳng được tạo bởi điểm thứ nhất nối với điểm thứ hai,thứ hai nối với thứ ba, thứ ba nối với thứ tư, các điểm ảnh được xác định trong thẻ Coordinate. Một ví dụ khác coordlndex [0 1 -1 2 3] ở dãy kí hiệu -1 cho biết hình ảnh được tạo trong VRML gồm hai đường thẳng: một đường thẳng tạo bởi điểm thứ nhất nối với điểm thứ hai và một đường thẳng khác tạo bởi điểm thứ ba nối với thứ tư.
* Colorlndex: được sử dụng khi colorPerVexter nhận giá trị FALSE, khi colorPerlndex nhận giá trị TRƯE thì có thể bỏ qua trường colorlNdex trong thẻ IndexedLineSet bởi vì nếu colorPerlndex nhận giá trị TRƯE thì bắt buộc phải thiết lập mầu cho mỗi điểm ảnh.
* ColorPerVexter: có giá trị boolean. Khi colorPerVexter nhận giá trị TRƯE thì màu của đườn thẳng sẽ là màu trung bình cả hai màu tại hai điểm ảnh tạo nên đường thẳng.

Cú pháp

IndexedLineSet{

Coord coordinate[] coordlndex

color Color[] colorIndex[] colorPerVexter TRUE

}

Ví dụ

Shape{

Appearance Appearance{}

Geometry IndexedLine s et {

Coord Coordinate{

Point[0.0 0.0 0.0,0.0 4.0 0.0, 2.0 0.0 0.0,]

}

Color Color{

Color[1.0 1.0 1.0,1.0 0.0 0.0,0.0 0.0 1.0 ]

}

coordIndex[0 1 2 0]

colorPerV exter TRUE

}

}

* Xây dựng khung bề mặt trong không gian

Thẻ IndexedFaceSet sẽ tạo lên bề mặt bằng cách kết họp các điểm với nhau.

Cú pháp:

IndexedFaceSet

{

Coord Color[] coord!ndex[]

color color[]

colorIndex[]

ColorPerVexter TRUE

convex TRUE

solid TRUE

}

* Coord, coordlndex, color, colorlndex, colorPerVexter có đặc điểm tương tự như đã nói ở trên. Chú ý là bề mặt luôn luôn được xác định bởi các đường thẳng khép kín vì thế không phải chỉ ra điểm đầu tiên lại một lần nữa trong trường coord.

Ví dụ

Coord Coordinate

{

Point [0 0 0,1 0 0,1 1 0,0 1 0]

}

Đã có bốn điểm xác định, khi liên kết các điểm này sẽ tạo ra hình vuông bằng cách sử dụng thuộc tính coordIndex[] và coordlndex[0 12 3] nói rằng điểm thứ nhất nối với điểm thứ hai, điểm thứ hai nối với điểm thứ ba, điểm thứ ba nối với điểm thứ tư và điểm thứ tư nối với điểm thứ nhất.

* Convex định nghĩa bề mặt là lồi hay lõm, trường này có giá trị kiểu boolean. Bộ trình duyệt VRML chỉ vẽ bề mặt lồi, trong trường hợp vẽ bề mặt lõm, bộ trình duyệt chia bề mặt đó thành các bề mặt lồi bé hơn để vẽ.

Ví dụ

Shape

{

Appearance appearance{}

Geometry IndexedFaceSet

{

Coord coordinate

{

Point[0 0 0,1 1 0,0 1 0,1 1 0,1 0 0,1 0 1]

coordlndex[0 123-145 6]

color Color{Color[l 1 1,1 0 0]}

colorlndex [0 1]

colorPerVexter FALSE

convex FALSE

solid FALSE

}

}

* Xây dựng khung lưổi và bản đồ trong không gian

Thẻ ElevationGrip cho phép xây dụng khung lưới được tạo bởi các điểm có độ cao xác định trong không gian. Thẻ này rất hữu ích cho việc xây dựng các mạng lưới hoặc địa hình.

Hình ảnh xây dụng được đặt ừong không gian của mặt phang OXYZ. Điểm bắt đầu là gốc tọa độ, các điểm còn lại tạo nên lưới phải nằm theo hướng dưong của các trục ox avf OZ.

Cú pháp:

ElevationGrid

{

xDimension 0 xSpacing 0.0 zDimension 0 zSpacing 0.0 heightn

color color[]

colorPerVexter TRUE

convex TRUE

solid TRUE

}

Tham số

* Color, colorPerVexter, convex, solid có tính chất tương tự như đã nói ở phàn trên.
* xDimension chứa số điểm bên trong lưới nằm trên trục X.
* zDimension chứa số điểm bên trong lưới nằm trên trục z.
* xSpacing là khoảng cách của hai điểm liên tiếp nhau theo hướng của trục X.
* zSpacing là khoảng cách của hai điểm liên tiếp nhau theo hướng của trục z.
* height chứa một danh sách các giá trị độ cao của mỗi điểm trong lưới. Các điểm này được tính theo thứ tự từ trái sang phải từ trên xuống dưới.

Ví dụ

Shape {

appearance Appearance { material Material { }

}

geometry ElevationGrid

{

xDimension 9

zDimension 9

xSpacing 1.0

zSpacing 1.0

solid FALSE

height [

0.0, 0.0, 0.5,1.0, 0.5, 0.0, 0.0, 0.0, 0.0,

0.0, 0.0, 0.0, 0.0, 2.5, 0.5, 0.0, 0.0, 0.0,

0.0, 0.0, 0.5, 0.5, 3.0, 1.0, 0.5, 0.0, 1.0,

0.0, 0.0, 0.5,2.0, 4.5, 2.5, 1.0, 1.5, 0.5,

1.0, 2.5, 3.0,4.5, 5.5, 3.5, 3.0, 1.0, 0.0,

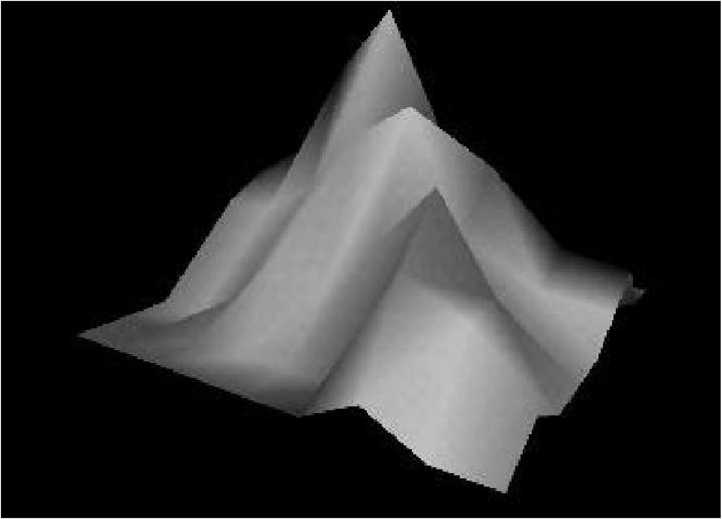
0.5, 2.0,2.0,2.5, 3.5, 4.0, 2.0, 0.5, 0.0,

0.0,0.0,0.5,1.5,1.0,2.0,3.0,1.5,0.0,

0.0,0.0,0.0,0.0,0.0,0.0,2.0,1.5,0.5,

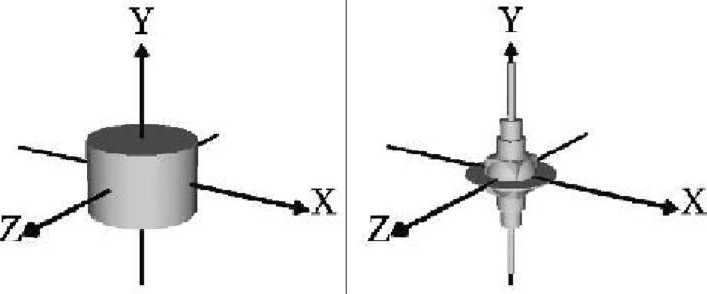
0.0,0.0,0.0,0.0,0.0,0.0, 0.5,0.0,0.0, ]}

}

Kết quả:

*Hình 2.1 Ví dụ về xây dựng khung lưới và bản đồ*

### Các phép biến đổi trong VRML

Một file VRML xây dựng các thành phần của thế giới ảo. Mặc đinh mọi hình ảnh được xây dựng trong VRML đều được đặt ở tọa độ gốc. Biến đổi là tạo ra một hệ thống hệ trục tọa độ mới mà hệ truc này có vị trí tương đối so với hệ tọa độ mộc định. Có các phép biến đổi như: di chuyển, co giãn, xoay.

*Hình 2.2 Trạc XYZ với 1 hình cơ bản và 1 hình phức tạp*

Nhóm Node Tranform quản lý các phép biển đồi tượng VRML như : dịch chuyển, co dãn, xoay. Node transform gồm có các node con: translatíon, rotation, scale, children...

* **Phép tịnh tiến ( Translation )**

Ví dụ:

Transform {

translation 2.0 0.0 0.0

children [... ]

}

Tham số :

translation x y z

* **Phép xoay ( Rotation )**

Ví dụ:

Tranform {

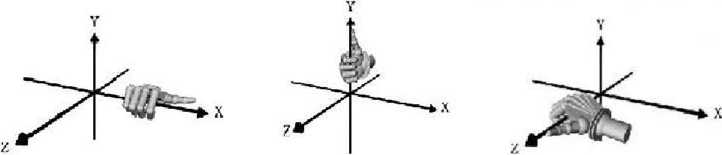
rotation 0.0 0.0 1.0 0.52

children [... ]

}

Tham số :

* rotation x y z Góc
* Góc = radian = degrees / 180.0 \* 3.141

Lưu ý: để dẽ dàng trong việc xoay đôi tượng theo đúng chiều ta có quy tắc bàn tay phải.

*Hình 2.3 Quay theo trục x ,y ,z*

Nắm bàn tay phải lại, choãi ngón cái ra theo như hình vẽ. Ngón tay cái chỉ phương của trục quay thì chiều từ cồ tay đến các ngón tay chi chiều dương khỉ quay đối tượng.

* **Scale** : co dãn đối tượng

Ví dụ:

Tranform {

scale 0.5 0.5 0.5

children [... ]

}

Tham số:

* scale x y z
* **Children** : Các đối tượng đước vẽ trong này sẽ chịu ảnh hưởng của các phép biến đổi node Tranform.

Vi dụ:

Tranform {

translation -2.0 -1.0 0.0

children [

Shape{

appearance Appearance { material Material {}

geometry Cylinder { radỉus 0.3 heỉght 6.0 top FALSE }

}

]

}

Shape {

Geometry Box { size 1.0 1.0 1.0}

}

Như ví dụ trên thì hình trụ sẽ dịch chuyển so với tọa độ gốc theo trục X là -2.0 và theo trục Y là -1.0, còn hĩnh hộp thì nằm tại tọa độ gốc

### Màu sắc trong VRML

Các đối tượng khi vẽ sẽ có mặc định là màu trắng và bạn có thể kiểm soát các thứ như : Shading color, Glow color, Transparency, Shininess, Ambient intensity. Màu sắc trong VRML được thể hiện qua 3 thông số là R G B( red- green-blue) với giá trị nằm trong khoảng từ 0.0 đến 1.0.

Node material nằm trong node appearance quản lý việc này. Node material có các node con là :

* diffuseColor: màu sắc khuyết tán.
* emissiveColor: màu sắc của ánh sáng phát ra từ đối tượng.
* transparency : độ trong suốt ( 0 : tối mờ, 1 : nét).
* ambientlntensity : xác định hàm lượng ánh sáng mà đối tượng phản chiếu.
* specularColor : cho biết màu sắc của các pixel trên bề mặt của đối tượng.
* shininess : cường độ sáng.

Ví dụ :

Shape {

appearance Appearance {

material Material {

diffiiseColor 0.8 0.8 0.8

emissiveColor 0.0 0.0 0.0

transparency 0.0

specularColor 0.71 0.70 0.56

shininess 0.16

ambientlntensity 0.4

}

}

geometry….

}

### Nhóm Node

Ta có thể nhóm các đối tượng đơn lẻ thành những đối tượng phức tạp. Việc đặt một nhóm này vào nhóm kia tạo thành một lược đồ cấu trúc các node. Một node trong nhóm có thể có bất kì các node con nào đặt bên trong trường children.

Có các node nhổm sau được hỗ trợ trong VRML: Anchor, Bỉllboard, Group, Switch, Tranform, Inline…

* **Anchor**: cho phép xác định một tập hợp các đối tượng và cách liên kết đến một url, chẳng hạn như là một siêu liên kết đến thế giới VRML khác, đến một trang HTML hoặc đến một dữ liệu nào đó mà toàn bộ trình duyệt có thể đọc được. Khi kích vào một trong các đối tượng bên trong node Anchor thì toàn bộ trình duyệt sẽ đưa đến địa chỉ url. Khi đưa chuột qua đối tượng nào đó chứa trong node thì có thể nhìn thấy được địa chỉ url của nó.

Ví dụ:

Anchor{

Children[

Shape{

Geometry Sphere{}}]

url “http://www.cuasoit.com”

description “Toi trang web cua lop CT 901”

}

Các thuộc tính:

* children: chứa tất cả các node bên trong nhóm
* url: cho biết địa chỉ url được tìm kiếm. Có thể có nhiều vị trí đặt trong url, khi đó bộ trình duyệt sẽ tìm kiếm dữ liệu trong các vị trí này theo thứ tự ưu tiên giảm dần.
* parameter: cung cấp thông tin thêm vào cho bộ trình duyệt. Cho ví dụ có thể chỉ rõ là cửa sổ cần xem mà url hiển thị vào đó.
* description: đưa ra một xâu, xâu này sẽ hiển thị khi người dùng di chuột qua đối tượng chứa trong node Anchor.
* bboxCenter: xác định trung tâm của một hình hộp chữ nhật mà nó bao quanh tất cả các node trong nhóm. Giá trị của trường này là một điểm trong không gian.
* bboxSize: xác định kích thước của hình hộp chữ nhật bao quanh tất cả các node trong nhóm. Mặc định trường này có giá trị -1 -1 -1 nghĩa là không có hình hộp nào. Tất cả các giá trị thành phần của trường này là lớn horn hoặc bằng O.Neu tất cả các node con không nằm trong hình hộp thì kết quả không xác định. Hai trường bboxCenter và bboxSize là tùy chọn.
* **Group**: tạo ra một tập hợp các vật thể như là một thực thể đơn. Node này có các trường: children, bboxCenter, bboxSỉze. Các trường này tương tự như của node Anchor. Tất cả các node con trong Group đều hiển thị.

Ví dụ:

Group {

Children[]

bboxCenter 0 0 0

bboxSize -1 -1 -1

}

* **Switch**: tạo ra một nhóm chuyển đổi. Chỉ có một node con trong nhóm được hiển thị. Và node này là do người dùng lựa chọn. Node con tuyệt đối được đánh số từ 0, không lựa chọn đánh là -1.

Ví dụ:

Các thuộc tính:

* whichChoice
* choice
* **Transform**: xác định hệ trục tọa độ mới cho các đối tượng trong cùng một nhóm. Node này đã được trình bày ở trên.
* **Billboard**: node này tạo ra một nhóm với hệ tọa độ dặc biệt. Tất cả các node trong nhóm đều được hiển thị. Hệ tọa độ này luôn được quay về đối diện với người dùng.

Vi dụ:

Billboard{

axisOfRotation 0.0 1.0 0.0 children [....]

}

Các thuộc tính:

* axisOfRotationbboxCenter
* bboxSize
* Inline: node này sẽ được trình bày ở phần sau

### Một số phương pháp vẽ trong VRML

Trong VRML chúng ta có thể vẽ trực tiếp đối tượng. Tuy nhiên trong nhiều trường họp, đối tượng được sử dụng nhiều lần thì việc vẽ trực tiếp sẽ rất mất thời gian, công sức và tăng dung lượng flle lên đáng kể.

Có 2 cách thông dụng nhất để giải quyết việc này đó là:

* **Inline**: hàm Inline cho phép ta gọi trực tiếp một đối tượng bên ngoài fíle vào file hiện tại.

Cú pháp:

Inline {url ”Tên đường dẫn”}

Vi dụ:

Inline {url ”robo.wrl”}

* **DEF** ( Define ): cho phép ta định nghĩa một đối tượng hay một kiểu thuộc tính

Vi dụ:

Shape {

appearance Appearance {

material DEF RedColor Material { diffiiseColor 1.0 0.0 0.0

}

}

geometry. . .

}

Shape {

appearance Appearance { material USE RedColor

}

geometry. . .

}

Như ví dụ trên ta định nghĩa đối tượng RedColor kiểu Material. Sau đó muốn sử dụng lại ta gọi hàm bằng từ khóa USE.

### Texture Mapping

Ta có thể mô hình mỗi chi tiết kết cấu nhỏ bé của một đối tượng. Thay vì phải mất thời gian để vẽ ta có thể lấy một bức ảnh dán lên bề mặt kết cấu như là dán decal. Kĩ thuật này được gọi là ánh xạ kết cấu.

Trong VRML có thể sử dụng một trong các định dạng ảnh sau đây:

* GIF: 8bit, có hỗ trợ trong suốt, không là sự lựa chon tốt cho texture mapping.
* JPEG: từ 8-16 bit, không hỗ trợ trong suốt, là sự lựa chọn tốt cho texture mapping.
* PNG: từ 8-16 bit, hỗ trợ trong suốt trên từng điểm ảnh, là sự lựa chon tốt nhất cho texture mapping.

### Script

Thẻ script đưa ra khả năng linh hoạt, mềm dẻo để mở rộng gần như vô hạn các yêu cầu hành động của chúng ta cũng như hành động mong đợi phản hồi từ thế giới ảo của chúng ta. Thẻ script cho phép chúng ta xác định cách tương tác với thế giới ảo bằng việc sử dụng các ngôn ngữ lập trình như Java, Javascript hoặc VRML script mới đây được đưa ra bởi Silicon Graphics, Inc.

Trong một thể Script có thể có các trường hay các thuộc tính, nhung không giống với các thẻ khác, trong thẻ script chúng ta có thể xác định các sự kiện mà thẻ này có thể nhận và gửi sự kiện. Một thẻ script sẽ xác định một hành động nào đó mỗi lúc nó nhận một sự kiện, trong mỗi hành động thẻ script có thể tạo ra nhiều sự kiện. Hơn nữa thẻ script có thể được dụng để xây dụng các thủ tục khởi tạo (initiazation procedures) và dừng, tắt (shutdown procedures).

Một thẻ script có thể nhận bất kỳ các sự kiện, các giá trị mà các sự kiện này có được là các biến thuộc loại eventln. Tên của các biến này là tên của các sự kiện mà thẻ script nhận sự kiện đó. Các biến này là chỉ có thể đọc nghĩa là chúng ta không thể thiết lập giá trị trực tiếp cho bất kỳ biến nào thuộc loại này. Một thẻ script cũng tạo ra bất kỳ các sự kiện nào, các giá trị mà các sự kiện này có được là các biến thuộc loại eventOut.

Cú pháp

Script

{

url[]

directOutput FALSE

mustEvahiate FALSE

eventln Datatype EventName

eventOut Datatype EventName

field Datatype FieldName InitialValue

}

* url cho biết chương trình script được xử lí. Các ngôn ngữ gần đây hỗ trợ cho việc xây dựng các chương trình script này là: JAVA, JavaScript, VRMLScript.
* Có hai cách mà có thể thay đổi thế giới ảo bằng việc sử dụng script: thứ nhất là sử dụng một biến eventOut và đặt sự kiện đến thẻ khác, thứ hai là truy cập trực tiếp các trường của thẻ khác, chú ý là cả hai cách này đều phải thực hiện trong ther script.
* Điều khó khăn cho bộ trình duyệt là vừa phải vẽ thế giới ảo vừa phải xử lý các sự kiện ở cùng một thời gian.

# Ứng Dụng VRML trong việc xây dựng ký túc xá ĐH CNHN

## Bài toán

## Yêu cầu đặt ra và hướng giải quyết

### Xây dựng ký túc xá

### Trang trí ký túc xá

### Xây dựng các đối tượng tỏng ký túc xá

### Di chuyển bên trong ký túc xá

### Chọn đối tượng

### Ánh sáng

### Đóng mở cửa

### Tối ưu khung hình

## Kết quả đạt được và hướng phát triển

### Kết quả