

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

-----🙞🙜🕮🙞🙜-----



**BÁO CÁO BÀI TẬP LỚN**

**CÔNG NGHỆ THỰC TẠI ẢO**

Đề tài: **Mô phỏng công trường xây dựng**

Giáo viên hướng dẫn: **Vũ Đức Huy**

Nhóm thực hiện: Nhóm 16

Lớp: KHMT 4 – K10

Thành viên trong nhóm:

1. Lê Quang Trung
2. Nguyễn Phùng Hải Chung

**Hà Nội. Ngày 13 tháng 6 năm 2018**

Mục lục

[LỜI NÓI ĐẦU 3](#_Toc516685482)

[CHƯƠNG I: TỔNG QUAN VỀ MÔN HỌC VÀ ĐỀ TÀI 5](#_Toc516685483)

[I. Khái niệm thực tại ảo là gì? 5](#_Toc516685484)

[II. Ứng dụng của công nghệ thực tại ảo 5](#_Toc516685485)

[CHƯƠNG II: TỔNG QUAN VỀ NGÔN NGỮ MÔ HÌNH THỰC TẠI ẢO – VRML 7](#_Toc516685486)

[(Vitual Real Model Language) 7](#_Toc516685487)

[I. Khái niệm VRML 7](#_Toc516685488)

[II. Các đối tượng hình học cơ bản trong VRML 8](#_Toc516685489)

[III. Các nút trong VRML 10](#_Toc516685490)

[IV. Các nút nội suy trong VRML 13](#_Toc516685491)

[V. Các phép biến đổi trong VRML 14](#_Toc516685492)

[VI. Các sự kiện trong VRML 16](#_Toc516685493)

[VII. Các ROUTER 16](#_Toc516685494)

[VIII.Tái sử dụng trong VRML 17](#_Toc516685495)

[CHƯƠNG III: ÁP DỤNG VRML MÔ PHỎNG CÔNG TRÌNH XÂY DỰNG 20](#_Toc516685496)

[I. Khảo sát hệ thống thực 20](#_Toc516685497)

[II. Kết quả đạt được 21](#_Toc516685503)

[KẾT LUẬN 28](#_Toc516685504)

**LỜI NÓI ĐẦU**

 Trong thời đại khoa học công nghệ phát triển, có rất nhiều phầm mềm, ứng dụng được xây dựng nhằm phục vụ nhu cầu của con người. Trong đó không thể không kể tới ứng dụng VRML, giúp con người mô phỏng các sự vật một cách chân thực để người xem có thể thấy được các sự vật 1 cách trực quan nhất mà không cần mẫu thật.

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.

Chúng em xin chân thành cảm ơn thầy Vũ Đức Huy đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình làm bài tập lớn.

**CHƯƠNG I: TỔNG QUAN VỀ MÔN HỌC VÀ ĐỀ TÀI**

1. **Khái niệm thực tại ảo là gì?**

Thực tại ảo (Virtual reality – VR) là thuật ngữ mô tả một môi trường được tạo mô phỏng bằng máy tính. Trong đó đồ họa máy tính được sử dụng để đảm báo yếu tố thực của môi trường này.

Các thành phần trong môi trường ảo đó có thể thay đổi theo các tác động của người sử dụng qua các hành động, lời nói, cái nhìn thậm chí là suy nghĩ. Đó chính là đặc tính chính của thực tại ảo: 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 các thành phần trong thế giới ảo. Người sử dụng nhìn thấy sự vật thay đổi trên màn hình mô phỏng ngay theo ý muốn của họ và bị thu hút bởi sự mô phỏng này. Sự thu hút này được tạo ra bởi những thay đổi trong môi trường ảo đã tác động lên các cơ quan cảm giác của con người thông qua một số thiết bị hiển thị cũng như tương tác đa dạng như: thị giác (từ các hình ảnh 3D ), thính giác (qua các âm thanh), xúc giác.

Có thể nói, VR- Thực Tại 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 như thị giác, thính giác, xúc giác.

1. **Ứng dụng của công nghệ thực tại ảo**

Ngày nay, tại nhiều nước phát triển, thực tại ảo đã và đang được ứng dụng trong rất nhiều lĩnh vực: khoa học kỹ thuật, kiến trúc, quân sự, giải trí,... và đáp ứng mọi nhu cầu: nghiên cứu - giáo dục - thương mại. Lĩnh vực ứng dụng mang lại nhiều hiệu quả nhất của thực tại ảo đầu tiên phải kể đến là y học. Bên cạnh đó thực tại ảo cũng được ứng dụng trong giáo dục, nghệ thuật, giải trí. Trong lĩnh vực quân sự, thực tại ảo 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 trong thời gian gần đây của thực tại ảo như: thực tại ảo ứng dụng trong sản xuất, thực tại ảo ứng dụng trong ngành rôbốt, thực tại ảo ứng dụng trong hiển thị thông tin (thăm dò dầu mỏ, hiển thị thông tin khối,....) thực tại ảo có tiềm năng ứng dụng vô cùng lớn. Có 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.

**CHƯƠNG II: TỔNG QUAN VỀ NGÔN NGỮ MÔ HÌNH THỰC TẠI ẢO – VRML**

(Vitual Real Model Language)

1. **Khái niệm VRML**

VRML (Virtual Reality Modeling Language) là ngôn ngữ mô hình hóa thực tại 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. VRML còn được dự trù trở thành một chuẩn trao đổi đa năng cho đồ họa ba chiều tích hợp 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.

Về căn bản VRML chỉ đơn giản là một định dạng trao đổi 3D. Nó định nghĩa được các vấn đề thường được sử dụng trong các ứng dụng 3D, chẳng hạn như các biến đổi phân cấp (hierarchical tranformations), nguồn ánh sáng (light sources), điểm nhìn (view points), hình học (geometry), sương mù (fog), thuộc tính của chất liệu (material properties) và các bản đồ kết cấu (texture mapping).

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 quang cảnh trên Web. Các file 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 file 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 truyền 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 đơn giản, hỗ trợ dịch vụ web3D.

1. **Các đối tượng hình học cơ bản trong VRML**

Các đối tượng hình học trong thế giới ảo thường được cấu tạo từ các đối tượng hình học cơ bản như hình hộp, hình tròn, hình trụ, hình cầu … Và VRML cung cấp sẵn cho chúng ta một số các đối tượng hình học đơn giản đó dưới dạng các nút như **Box** (hình hộp), **Cone** (hình nón), **Cylinder** (hình trụ tròn), **Sphere** (hình cầu)… **Event Architecture – Kiến trúc sự kiện**

1. **Box- Hình hộp chữ nhật.**

Hình hộp chữ nhật tạo ra từ node Box có tâm tại tọa độ ( 0,0,0) và có kích

thước được quy định qua trường “size x y z” với x,y,z lần lượt là kích thước hình hộp theo các trục Ox, Oy, Oz trong hệ trục tọa độ Decac. Nếu để mặc định size của hình hộp (không khai báo giá trị của trường), thì hình hộp có kích thước mặc định từ -1 đến +1.

Shape {

geometry Box{ size 2.0 2.0 2.0 }

}

1. **Sphere – Hình cầu.**

Hình cầu được tạo ra từ node Sphere có tâm tại tọa độ (0,0,0) và có bán kính được quy định qua trường “radius”. Giá trị trường radius phải lớn hơn 0.0.

Shape {

geometry Sphere {

radius 1.0

}

}

1. **Cylinder – Hình trụ**

Hình trụ đuợc tạo ra từ node Cylinder có tâm tại tọa độ (0,0,0), có bán kính quy định qua trường radius và có chiều cao quy định qua trường height.

Ba trường còn lại trong node quy định sự tồn tại (nếu mang giá trị TRUE) hoặc không tồn tại (khi mang giá trị FALSE): tương ứng là mặt đáy trên – top, mặt đáy dưới – bottom, và mặt bên – side.

Các mặt không tồn tại sẽ không được xét tới nếu toàn khối trụ tham gia vào các sự kiện (ví dụ như phát hiện ra va chạm hoặc kích hoạt cảm biến).

Shape {

geometry Cylinder{

height 2.0

radius 1.0

bottom TRUE

top TRUE

side TRUE }

}

1. **Cone – Hình nón.**

Hình nón được tạo ra từ node Cone có tâm tại (0,0,0), có bán kính mặt đáy quy định qua trường bottomRadius và có chiều cao quy định qua trường height.

Shape {

geometry Cone {

height 2.0

bottomRadius 1.0

bottom TRUE

side TRUE }

}

1. **Các nút trong VRML**

VRML bao gồm 54 nút khác nhau và được phân loại làm 9 nhóm chính dựa trên chức năng và các hàm của các nút. Bao gồm:

Grouping Nodes: Nhóm các nút nhóm.

* Anchor
* Billboard
* Collision
* Group
* Transform.

Special Groups Nodes: Nhóm các nút nhóm đặc biệt.

* Inline
* LOD
* Switch.

Sensors Nodes: Nhóm các nút cảm biến.

* CylinderSensor
* PlaneSensor
* ProximitySensor
* SphereSensor
* TimeSensor
* TouchSensor
* VisibilitySensor.

Geometry Nodes: Nhóm các nút đối tượng hình học.

* Box
* Cone
* Cylinder
* ElevationGrid
* Extrusion
* IndexedFaceSet
* IndexedLineSet
* PointSet
* Sphere
* Text.

Geometry Properties Nodes: Nhóm các nút thuộc tính hình học.

* Color
* Coordinate
* Normal
* TextureCoordinate.

Appearance Nodes: Nhóm các nút mô tả hiển thị.

* Appearance
* FontStyle
* ImageTexture
* Material
* MovieTexture
* PixelTexture
* TextureTransform.

Interpolators Nodes: Nhóm các nút nội suy.

* ColorInterpolator
* CoordinateInterpolator
* NormalInterpolator
* OrientationInterpolator
* PositionInterpolator
* ScalarInterpolator

Bindable Nodes: Nhóm các nút có thể ghép được

* Background
* Fog
* NavigationInfo
* Viewpoint.

1. **Các nút nội suy trong VRML**

Các nút thuộc nhóm nút **Interpolators**có chức năng giữ các giá trị xen vào các trường khi có sự kiện thay đổi giá trị các trường xảy ra (trừ các trường có giá trị kiểu logic). Các nút thuộc nhóm nút này có cú pháp giống nhau (có cùng các trường) chỉ khác nhau về kiểu dữ liệu.

Trong VRML cung cấp sẵn các nút **Interpolators** sau:

* ColorInterpolator
* CoordinateInterpolator
* NormalInterpolator
* OrientationInterpolator
* PositionInterpolator
* ScalarInterpolator

Mỗi nút đều có cú pháp như sau:

…Interpolator {

key […]

keyValue […]

}

Trong đó **key** là tập các giá trị đầu vào còn **keyValue** là tập các giá trị đầu ra. Trong các nút **Interpolators** đều có một sự kiện đầu vào là **set\_fraction** và sự kiện đầu ra là **value\_changed**, hai sự kiện này được nối với nhau tức là khi các nút này khi nhận được một sự kiện thì nó cũng tạo ra một sự kiện. Sự kiện set\_fraction xác định một giá trị **key** và sự kiện đầu ra xác định một **keyValue** tương ứng với giá trị **key**.

1. **Các phép biến đổi trong VRML**

Để tạo nên thế giới sử dụng tất cả mọi thứ, chúng ta cần có thể biến đổi được mọi đối tượng. VRML có 3 loại biến đổi có thể áp dụng cho các đối tượng. Đó là translation, rotations và scales và chúng được sử dụng trong nút transform, Nhưng không phải một nút transform chỉ có phép quay rotation. Các phép biến đổi trong transform có thế áp dụng cho các nút con children của nút. Điều này gọi là một tổ hợp, một nút cha có thé có nhiều nút con. Cú pháp cho vấn đề này được chỉ ra dưới đây, áp dụng cho nút transform

Transform{

Translation 1 1 1

Rotation 0 1 0 0.78

Scale 2 1 2

Children [

USE FBOX

]

}

Một nút transform có thể có các tổ hợp khác bên trong nút con của nó, cho phép bạn thực hiện tuần tự một chuỗi các phép biến đổi. Chú ý rằng trật tự các phép biến đổi là không quan trọng. Một phép biến đổi thực hiện sau phép quay. Bên trong một nút transform đơn lẻ, các phép biến đổi được thực hiện theo một trật tự chặt chẽ: Scale, rotation và translate. Vì vậy nếu bạn muốn thực hiện phép biến đổi sau phép quay, bạn cần tổ hợp các nút transform bên trong các nút khác.

1. **Translation and Scale.**

Có 2 phép biến đổi tương tự nhau, cả hai đều nhận ba đối số: giá trị x, y và z.

Phép biến đổi dịch chuyển trung tâm của đổi tượng trong các khoảng cách này theo hướng thích hợp. Tỉ lệ nhân với kích thước của đối tượng bởi các giá trị này theo hướng thích hợp. Một phép biến đổi 0 theo một hướng sẽ loại bỏ các đối tượng không ảnh hưởng đến hướng. Yếu tố tỉ lệ 0 làm cho đối tượng không xác định theo hướng đó là điều bình thường không được mong muốn. Yếu tố 1 yêu cầu mà không có ảnh hưởng nào cả.

Quan trọng chú ý rằng việc tính tỉ lệ là tương đối cho nguyên bản, không phải là trung tâm của đối tượng. Vì vậy để tính tỉ lệ của trung tâm đối tượng, chúng ta cần phải đảm bảo rằng đối tượng được đặt trung tâm tại phần mở đầu. Đó là lý do tại sao việc tính tỉ lệ phải thực hiện trước phép quay và phép biến đổi.

1. **Rotation**

Phép quay có điểm hơi khác so với hai loại ở trên. Nó nhận 4 tham số, đầu tiên là 3 tọa độ xác định trục của phép quay và cuối cùng là góc quay tính bằng radian. Ví dụ để quay 1 radian quanh trục Y bạn phải viết:

Transform{

Rotate 0 1 0 1

Childen [

USE FBOX

]

}

Chiều dài của trục quay bất kỳ, không cần thiết là 1. Bạn có thể sử dụng giá trị Y là 50 nếu bạn thích nhưng nó không thực hiện điều gì khác với giá trị Y là 1. Trục quay hoàn toàn là tùy ý, bạn có thể quay quanh bất kỳ trục nào bạn thích. Ví dụ trục 1 0.3 2.45 là hoàn toàn hợp lý. Thật là khó khăn để tạo các phép quay theo cách này, tuy nhiên chúng ta có thể khó khăn để quan sát, đặc biệt nếu bạn không có bất kỳ luyện tập nào. Chính vì điều đó Vapour Technology đã tạo ra một cặp công cụ Dizzy và Twister. Twister tạo phép quay vuông góc với trục từ một laoij đơn giản hơn và Dizzy kết hợp nhân các phép quay vuông góc với trục để tạo một loại đơn giản hơn với hiệu quả tương tự. Bạn có thể sử dụng các điều này để dễ dàng hơn trong việc quay các đối tượng trong thế giới.

1. **Các sự kiện trong VRML**

Một số trường hợp, hầu như các nút đều có chứa các sự kiện. Có hai loại sự kiện: eventIn và eventOut. Các eventOut là các sự kiện sắp đi ra, phát sinh các thông tin như thay đổi 1 giá trị hoặc thời gian kích chuột. Các eventIn là các sự kiện sắp đến, chấp nhận thông tin từ bên ngoài của nút và thực hiện một vài điểu với nó. Một sự kiện có một loại dữ liệu thích hợp với chúng.

Một vài nút có các trường được trưng bày. Điều này có nghĩa là nút có hai trường định nghĩa cho trường đó set\_fieldname và fieldname\_changed. Có eventIn và eventOut đối với trường có thể sử dụng để thiết lập các giá trị của nó và thông báo cho thế giói bên ngoài có khi có sự thay đổi. Nếu bạn sử dụng set\_fieldname để thiết lập giá trị của trường, nút sẽ phát sinh một sự kiện fieldname\_changed. Để dễ dàng sử dụng, các thành phần set\_ và \_changed của sự kiện có thể nghiêng về phía trái và browser sẽ làm việc khi sự kiện được sử dụng. Nếu một trường không được phơi bày ra, nó không thể thay đổi do các sự kiện và giá trị trong trường được sử dụng trong suốt các khoảng thời gian. Để xem xét trường nào được bày ra đối với mỗi nút, hãy giữ lấy một tham chiếu từ một vị trí từ trang liên kết hoặc nhận lấy tham chiếu nút trong phụ lục.

1. **Các ROUTER**

Để thực hiện hiệu quả mọi thứ với các sự kiện, chúng ta cần nối chúng với nhau. Việc kết nối được biết đến như ROUTER. Ví dụ để dẫn từ touchTime eventOut tới startTime eventIn, chúng ta sẽ dẫn sự kiện như sau:

ROUTE SSENSOR.touchTime TO SOUND.startTime

Vì vậy, bít này của mã sẽ dẫn tới sự kiện touchTime từ TouchSensor (sẽ đề cập tới ở phần sau) tới sự kiện startTime trong nút sound (cũng được để cập ở phần sau). Do đó khi TouchSensor bị kích vào âm thanh được phát ra. Bạn cần sử dụng DEF cho mỗi nút mà bạn dẫn tới hoặc từ đó sao cho nó có một tên riêng lẻ. Vì vậy các nút TouchSensor và Sound sẽ được định nghĩa:

DEF SENSOR TouchSensor{

}

DEF SOUND Sound {

}

Ngoại trừ các trường hợp bên trong chúng. Nếu bạn có một số các đối tượng với cùng một tên (do sử dụng USE), và dẫn tới hoặc từ chúng, tất cả các đối tượng bị ảnh hưởng, vì vậy nếu bạn chỉ muốn một đối tượng bị ảnh hưởng, đưa cho nó một tên duy nhất hoặc sử dụng PROTO (được giải thích sau).

1. **Tái sử dụng trong VRML**

Nếu bạn có nhiều đối tượng giống hệt nhau, thường rất khó khăn để duy trì việc viết chính xác nhiều đối tượng theo cùng một loại. Do đó, bạn có thể sử dụng các định nghĩa ở phần trước. Sử dụng hộp này, bạn có thể định nghĩa nó để có tên FBOX. Mỗi khi bạn muốn sử dụng lại hộp, bạn có thể chỉ gõ USE FBOX thay cho toàn bộ định nghĩa. Ví dụ:

DEF FBOX Shape {

apprearance Apprearance {

material Material {

}

}

geometry Box {

}

}

USE FBOX

Đây không chỉ làm một ví dụ đơn giản, khi tạo hai hộp chính xác tại cùng một vị trí, không chỉ là các thứ mà bạn muốn. Thỉnh thoảng, là định nghĩa USE tại một vài nơi có ý nghĩa hơn. Đồng thời bạn có thể DEF/USE bất kỳ loại nút nào vì vậy bạn muốn sử dụng lại Appearance của một đối tượng, bạn có thể thực hiện tốt điểu đó.

Shape {

appearance DEF APP1 Appearance {

material Material {

}

}

geometry Box {

}

}

Shape {

appearance USE APP1

geometry Box {

}

}

Một lần nữa bạn có thể thấy ở đây ta có thể tạo hai hộp ở cùng một vị trí trong thế giới. Một cách khác để sử dụng lại mã VRML là sử dụng các nút Inline. Điều này cho phép nhận dữ liệu từ các file bên ngoài và chèn nó vào file của bạn. Vì vậy nếu bạn có mô hình chiếc ghế gọi là chair.wrl bạn có thể chèn nó vào quang cảnh của bạn:

Inline {

url “chair.wrl”

}

File bạn đang kết hợp theo các này phải là VRML hợp lệ, vì vậy nó phải là các header và có mọi thứ cần thiết cho một file VRML.

**CHƯƠNG III: ÁP DỤNG VRML MÔ PHỎNG CÔNG TRÌNH XÂY DỰNG**

1. **Khảo sát hệ thống thực**


5. 1. 1. **Bài Toán**

Việc Viêc thiết kê một thiết kế một công trường xây dựng thực sự là rất khó khăn, dựa vào bản thiết kế mà có thế khái quát lên được công trường đó thì lại càng khó khăn hơn nữa và không phải ai cũng có thể hiểu được bản thiết kế. Chính vì vậy nhóm em đã sử dụng ngôn ngữ VRML xây dựng một công trường xây dựng mô phỏng để mọi người có thể tham quan chiêm ngưỡng công trường xây dựng ngay tại máy tính của mình mà vẫn có cảm giác như thật vậy. Mô hình được thể hiện trên nền một trang web nên mọi người ở nhiều nơi có thể tham gia trực tiếp vào mô hình mà không cần phải cài đặt gì nhiều.

* + 1. **Yêu cầu đặt ra và hướng giải quyết**

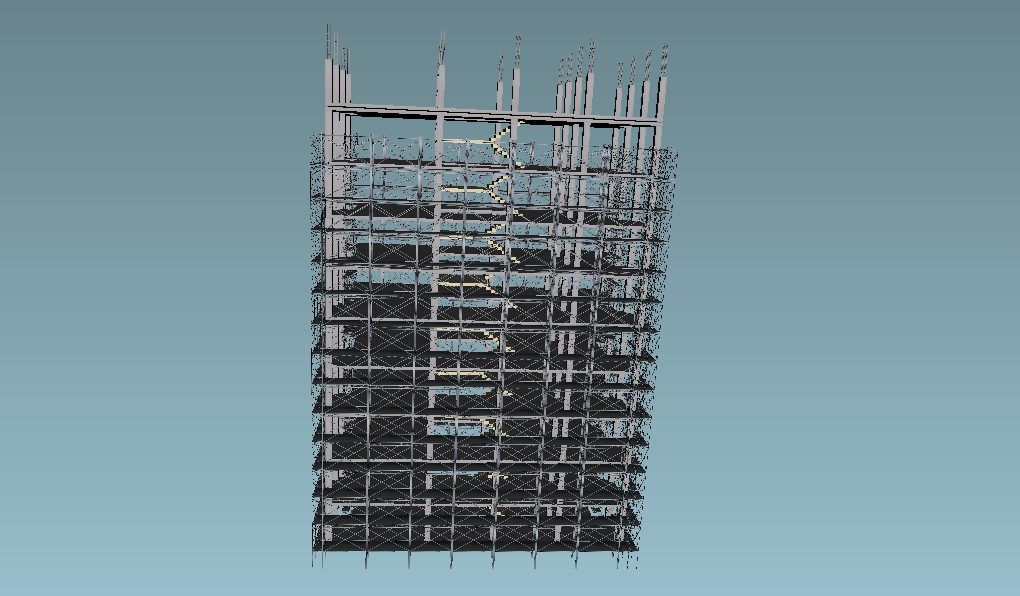
1. Về mặt thẩm mĩ

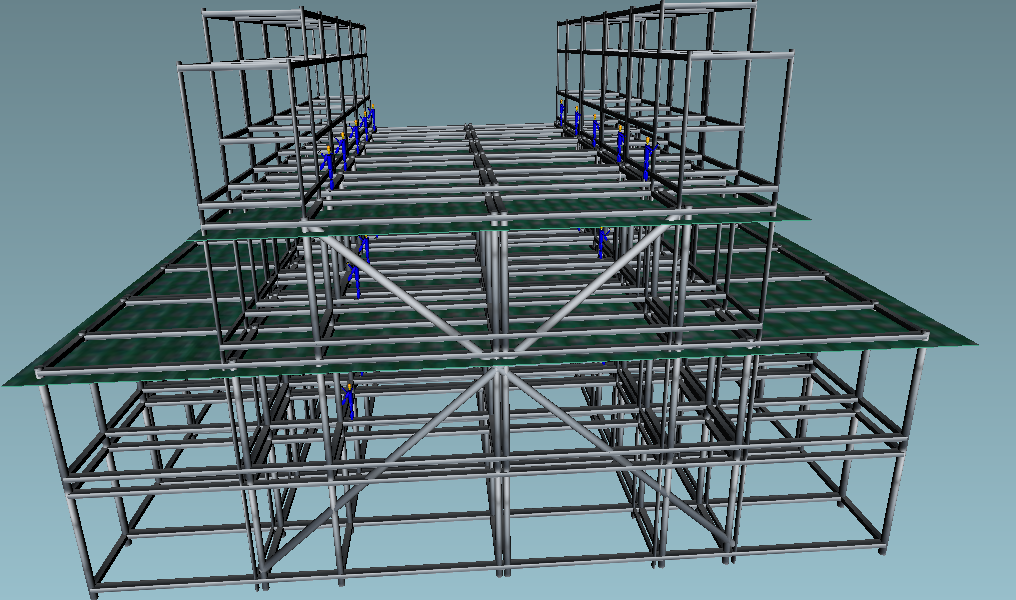
* Mô hình thể hiện được chân thực toàn cảnh công trường xây dựng.
* Kết câu và màu sắc như thật và bắt mắt

1. Về mặt kỹ thuật

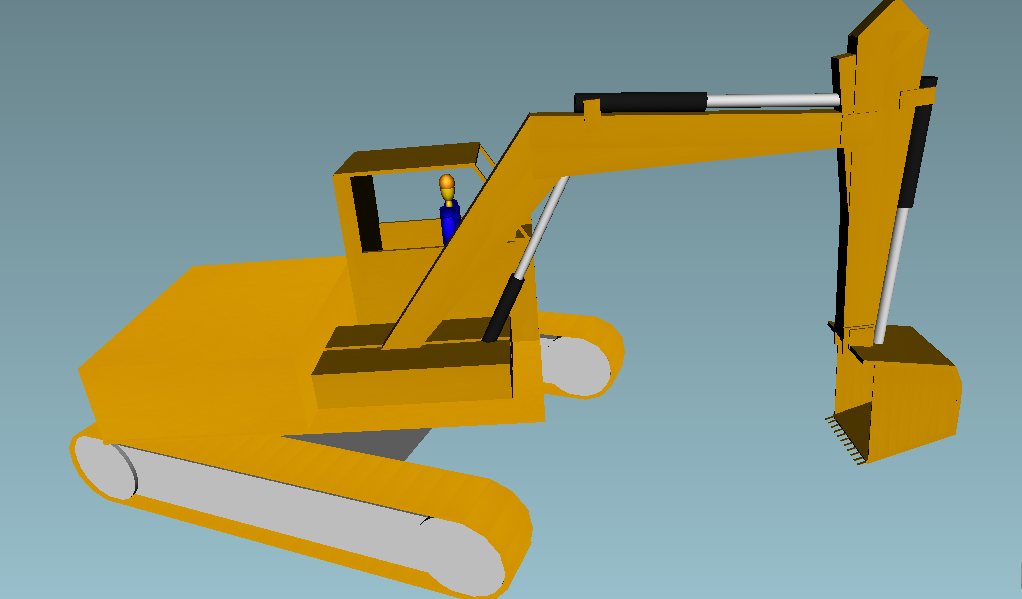
* Người dùng có thể tham gia trược tiếp quan sát toàn cảnh công trường xây dựng.
* Người dùng có thể tưởng tác như: mở đóng cửa,…
* Ứng dụng của mô hình gọn nhẹ, có thể truyền tải lên website một cách nhanh chóng

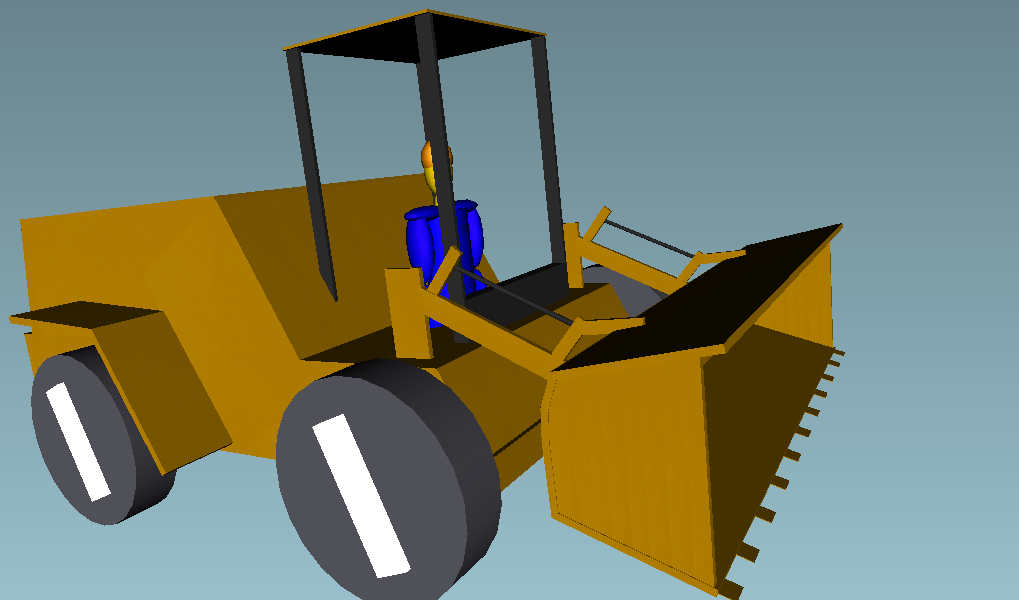
1. **Kết quả đạt được**
   * + 1. **Tòa nhà đang thi công**

****

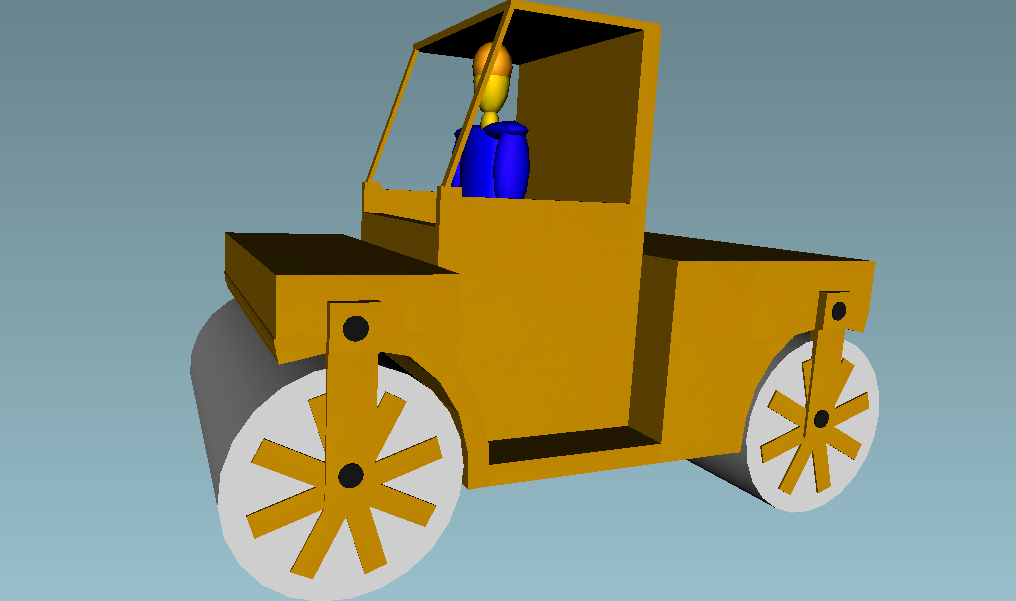
****

* + - 1. **Máy xúc**

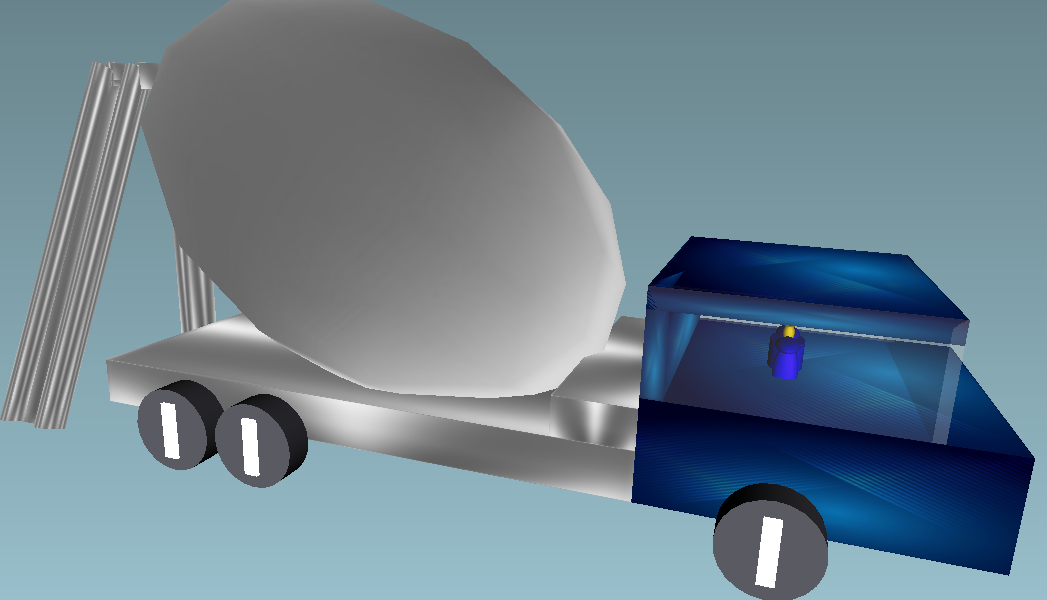
****

****

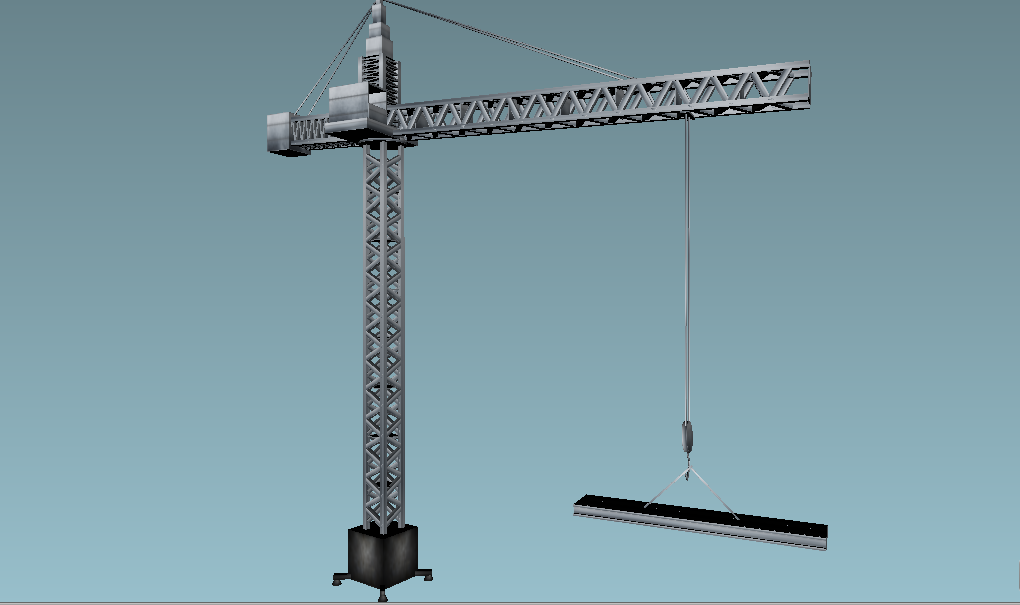
* + 1. **Xe lu**

****

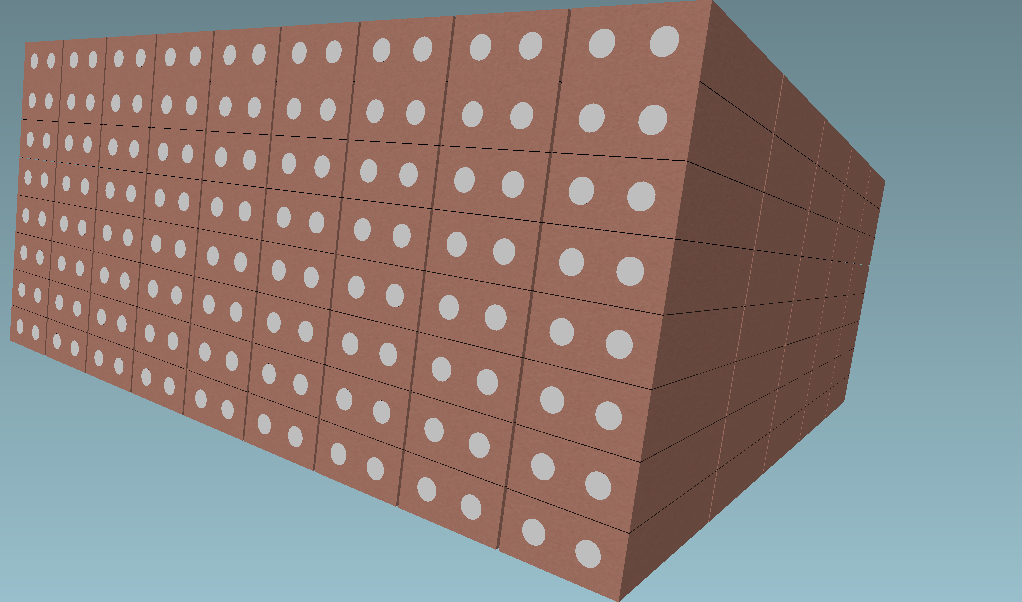
* + 1. **Xe trộn xi măng**

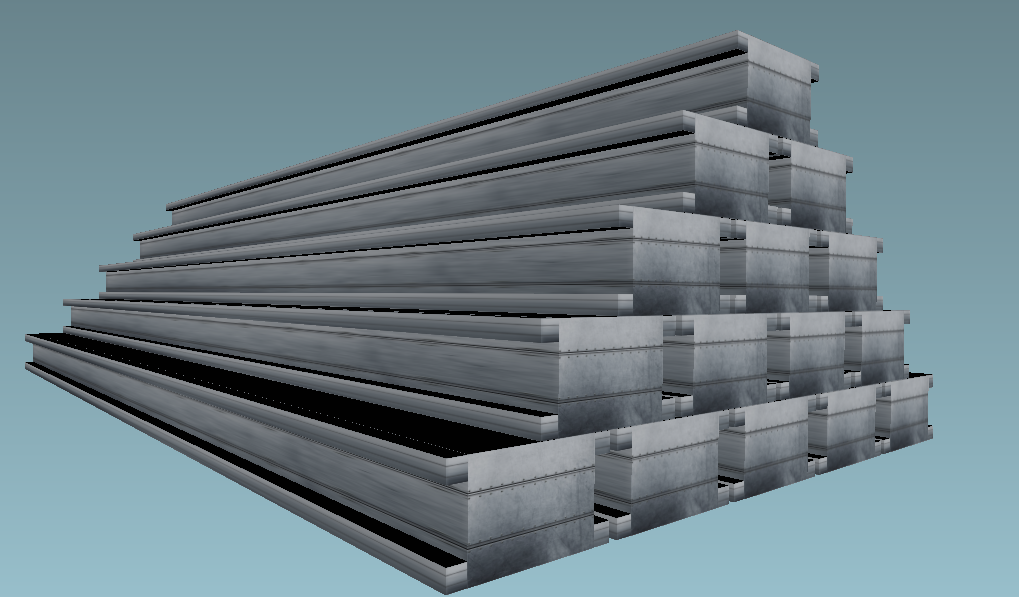
****

* + 1. **Cần cẩu**

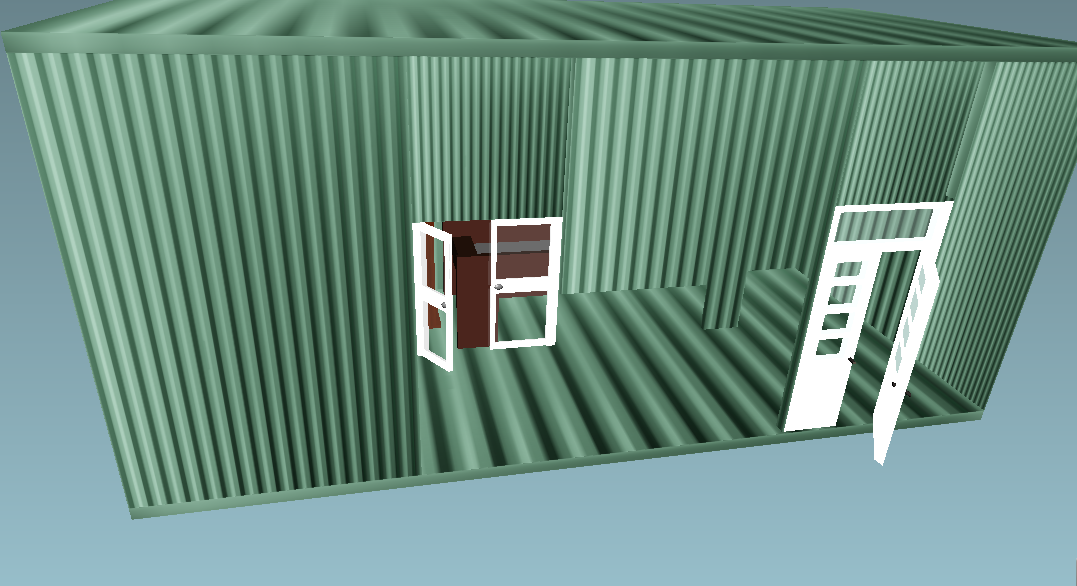
****

* + 1. **Một số vật liệu**

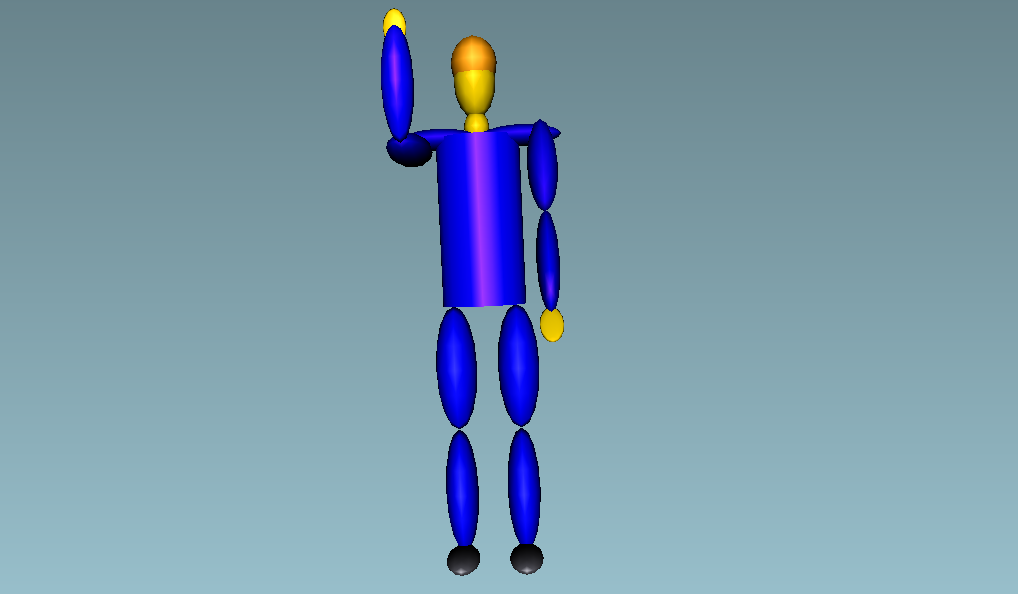
****

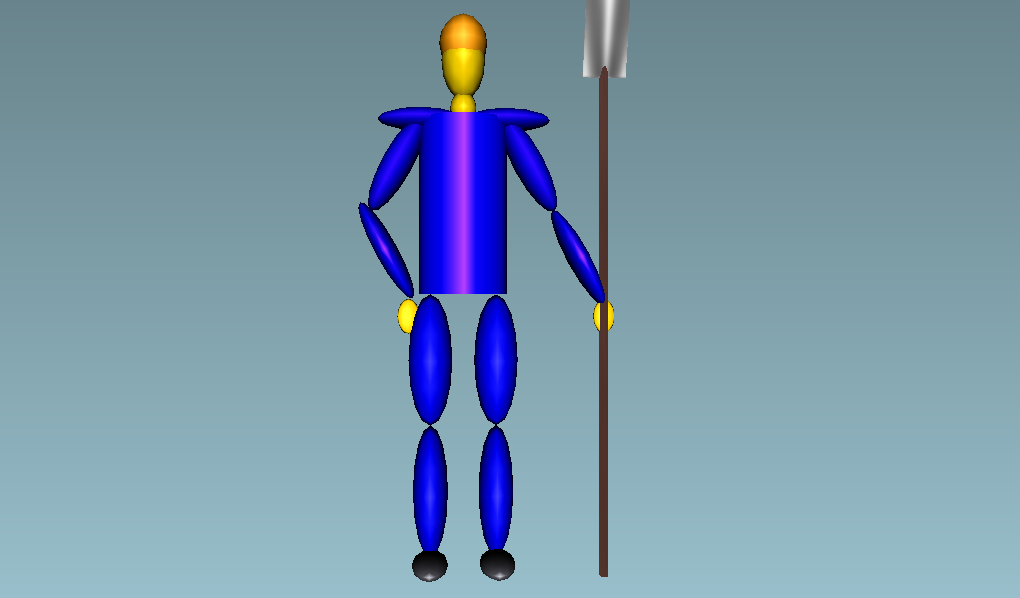
****

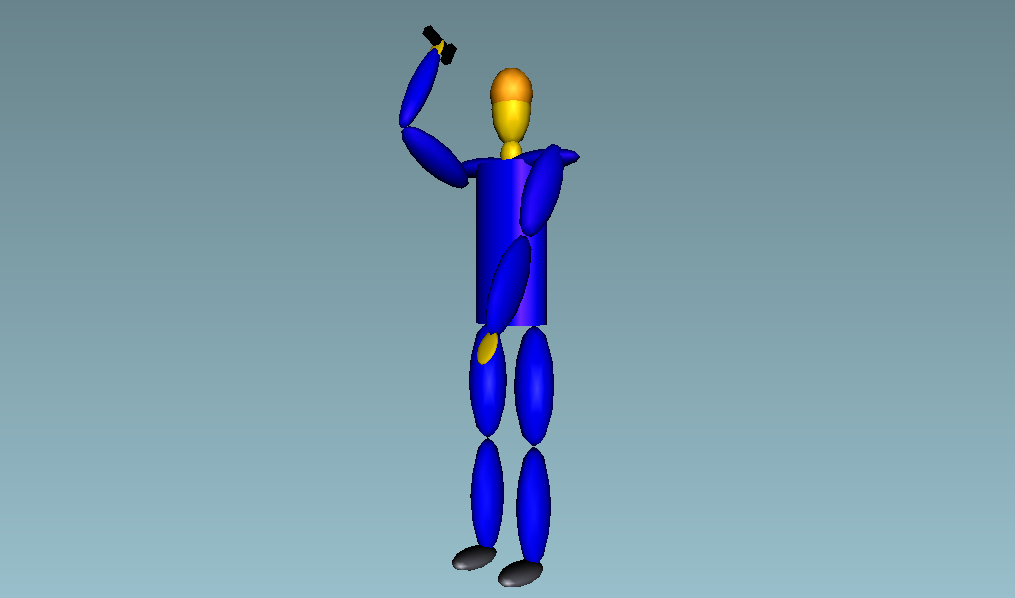
* + 1. **Container**

****

* + 1. **Người**

****

****

****

**KẾT LUẬN**

Qua thực hiện đề tài, kết quả đạt được:

- Nắm bắt được kiến thức các khái niệm cơ bản về công nghệ thực tại ảo, cũng như các ứng dụng quan trọng và rộng rãi của nó ngày nay.

- Nắm được kiến thức về ngôn ngữ vrml.

- Xây dựng thành công mô hình công trường xây dựng.

- Sản phẩm trong môi trường ảo khá giống và đồng nhất với thực tiễn.

- Thực hiện được các tương tác giữa người và đồ vật trong thế giới ảo.

- Củng cố khả năng tư duy lập trình cũng như cảm quan về đồ họa.

-Tăng cường khả năng làm việc theo nhóm.

Tuy đã cố gắng nỗ lực hết sức để thực hiện đề tài nhưng vì kiến thức và kinh nghiệm còn hạn hẹp, nên đề tài vẫn còn nhiều khuyết điểm thiếu sót chưa khắc phục được. Chúng em mong nhận được những lời nhận xét, góp ý của thầy giáo để hoàn thiện bài hơn.

Chúng em xin trân thành cảm ơn !