BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CNTT

------🙠🕮🙢------



**BÀI TẬP LỚN**

**MÔN HỌC: THỰC TẠI ẢO**

**ĐỀ TÀI**: **MÔ PHỎNG CÔNG TRƯỜNG LAO ĐỘNG**

**Giáo viên hướng dẫn : Th.s. VŨ MINH YẾN**

**Nhóm thực hiện: Nhóm 3 – Lớp: KHMT2 – K8**

1. **Phạm Văn Huệ 0841060155**
2. **Nguyễn Văn Đạt 0841060143**
3. **Lê Quốc Đạt 0841060165**

# LỜI MỞ ĐẦU

Trong xu thế tin học hóa toàn bộ cuộc sống đang diễn ra tấp lập hiện nay, lĩnh vực đồ họa đóng góp một vai trò cực kỳ to lớn. Ở đâu ta cũng có thể tìm thấy những minh họa cho việc áp dụng đồ họa máy tính vào cuộc sống: xử lý ảnh, phim hoạt hình, đặc biệt là thực tại ảo (Virtual Reality\_VR). Thực tại ảo là một ngành công nghệ cao, có ứng dụng đặc biệt rộng rãi. Từ các trò chơi 3D cực kỳ sống động đến những mô phỏng phức tạp trong công nghiệp và khoa học công nghệ(Mô Phỏng các vụ thử hạt nhân, mô phỏng thế giới ảo nhằm huấn luyện phi công…) từ những trường hợp đơn giản cho đến những trường hợp cực kỳ quan trọng.

Một phần quan trọng của VR là mô phỏng thế giới thực xung quanh chúng ta. Để làm được điều này, chúng ta có thể dùng ngôn ngữ VRML (Virtual Reality Modeling Language). Đây là một ngôn ngữ tương đối đơn giản nhưng cực kỳ hiệu quả. Với VRML , kết hợp với trình duyệt IE, chome, firefox, chúng ta có thể mô phỏng thế giới quanh ta một cách như ý. Trong thế giới đó các vật thể có các tương tác và các cảm biến y như thế giới thực. Ứng dụng của VRML cũng rất rộng rãi: mô phỏng nhà cửa, thiết kế các phòng học ảo, các thiết bị thí nghiệm ảo, lập trình không gian ảo,các khu vui chơi,mô phỏng đường phố…

Trong bài tập lớn này, chúng em đã dùng ngôn ngữ VRML để mô phỏng một phần của công trường lao động. Tuy được sự hướng dẫn tận tình của cô Vũ Minh Yến nhưng do thời gian và năng lực có hạn nên kết quả còn nhiều thiếu sót. Tuy nhiên chúng em rất cảm ơn cô vì môn học và bài tập lớn này đã đem lại cho chúng em nhiều hiểu biết về thế giới đồ họa, mở ra cho chúng em một hướng đi mới trong lĩnh vực công nghê thông tin.

# Mục Lục

[LỜI MỞ ĐẦU 2](#_Toc453220433)

[Mục Lục 3](#_Toc453220434)

[Chương 1: Giới thiệu VRML 4](#_Toc453220435)

[1.1. Khái niệm 4](#_Toc453220436)

[1.2. Lịch sử ra đời và phát triển của VRML 4](#_Toc453220437)

[1.3. Đặc điểm cơ bản của VRML 5](#_Toc453220438)

[1.4. Công cụ soạn thảo VRML 6](#_Toc453220439)

[Chương 2: Mô tả BTL và kết quả thực hiện 6](#_Toc453220440)

[2.1. Khảo sát các đối tượng 6](#_Toc453220441)

[2.2. Tòa Nhà 8](#_Toc453220442)

[2.3. Người công nhân 9](#_Toc453220443)

[2.4. Cần cẩu 10](#_Toc453220444)

[2.5. Máy ủi 13](#_Toc453220445)

[2.6. Bao xi măng 19](#_Toc453220446)

[2.7. Xe trộn bê tông 20](#_Toc453220447)

[2.8. Xe lu 20](#_Toc453220448)

[2.9. Xe rùa 21](#_Toc453220449)

[2.10. Công trường 25](#_Toc453220450)

# Giới thiệu VRML

## Khái niệm

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 thế giới thực và các đối tượng đồ họa tương tác ba chiều, sử dụng mô hình phân cấp trong việc thể hiện tương tác với các đối tượng của mô hình, được thiết kế dùng trong môi trường Internet, Intranet và các hệ thống máy khách cục bộ (local client) mà không phụ thuộc vào hệ điều hành.

Các ứng dụng 3D của VRML có thể truyền đi một cách dễ dàng trên mạng với kích thước khá nhỏ so với băng thông, phần lớn giới hạn trong khoảng 100 - 200KB. Nếu HTML là định dạng văn bản thì VRML là định dạng đối tượng 3D có thể tương tác và điều khiển thế giới ảo.

Hiện nay, VRML có lợi thế là sự đơn giản, hỗ trợ dịch vụ Web3D, có cấu trúc chặt chẽ, với khả năng mạnh mẽ, giúp cho việc xây dựng các ứng dụng đồ họa ba chiều một cách nhanh chóng và chân thực nhất.

VRML là một trong những chuẩn trao đổi đa năng cho đồ họa ba chiều tích hợp và truyền thông đa phương tiện, 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. Với mục đích xây dựng định dạng chuẩn cho phép mô tả thế giới thực trên máy tính và cho phép chạy trên môi trường web, VRML đã trở thành chuẩn ISO từ năm 1997.

## Lịch sử ra đời và phát triển của VRML

Năm 1994, lần đầu tiên VRML được thảo luận tại hội nghị WWW, Gieneva, Thụy Sĩ. Tim Berners-Lee và Dave Raggett đã tổ chức ra phiên họp có tên là Birds of a Feather (BOF) để mô tả giao diện thực tại 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 định 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-VRML được thành lập để tập trung vào xây dựng phiên bản VRML đầu tiên.

Vào tháng 3/ 1995, 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” gửi 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 tiêu chuẩn quốc tế (ISO) đã thống nhất ý kiến lấy phiên bản năm 1996 của VRML 2.0 để đưa ra xem xét vào tháng 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

Tiêu 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 trên 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.

VRML được thiết kế dành riêng cho việc hiển thị thế giới 3D và không phải là sự mở rộng của HTML.

## Công cụ soạn thảo VRML

* Sử dụng vrmlpad 3.0 để soạn thảo
* Để hiển thị các file VRML, sử dụng trình duyệt Cortona 3D Viewer của hãng Parallel Graphics. Phần mềm này sẽ giúp người dùng thuận tiện hơn khi xem các mô hình ảo trên máy tính một cách trực quan sinh động.

# Mô tả BTL và kết quả thực hiện

## Khảo sát các đối tượng

* Các đối tượng đươc khảo sát thông qua hình ảnh trên Google và quan sát trực tiếp.
* Một số hình ảnh minh họa:

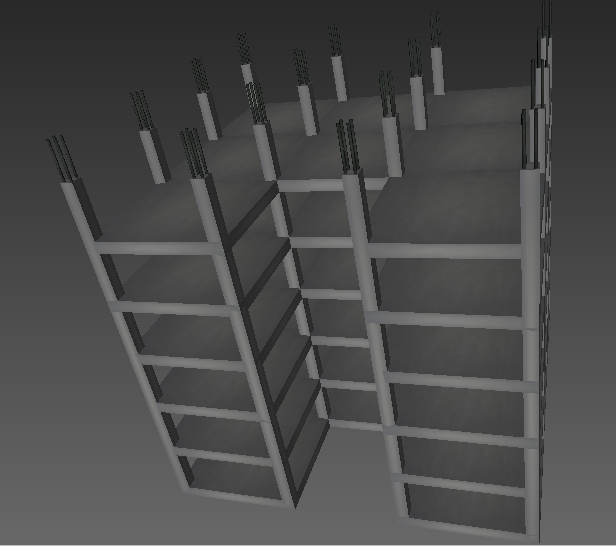






## Tòa Nhà

* Được xây dựng bằng cách ghép các dãy nhà với nhau.
* Mỗi dãy nhà được xây từ các hình hộp( box ) và hình trụ( xây dựng bằng Extrusion). Hình hộp làm sàn và cột, hình trụ làm cốt thép.



## Người công nhân

* Được xây dựng từ các bộ phận riêng biệt.
* Tay, chân, thân được xây dựng bằng Extrusion. Đầu, bàn tay, bàn chân, mũ bảo hiểm xây dựng bằng hìnhcầu (Sphere).
* Thân và mặt được dán ảnh để trông thật hơn.



## Cần cẩu

* Gồm 2 phầm chính là trụ và trục.
* Trụ xây dựng từ các hình hộp(box) được ghép nối với nhau bởi phép dịch chuyển (Translation), phép quay (Rotation) và tái sử dụng các đối tượng(Inline).
* Trục xây dựng từ các hình trụ (Cylinder) và hình được tạo bởi Extrusion được ghép nối với nhau bởi phép dịch chuyển (Translation), phép quay (Rotation) và tái sử dụng các đối tượng(DEF).
* Phần trục có thể quay được khi click chuột bởi nút nội xuy OrientationInterpolator và nhúng mã(Script).
* Code phần tác động quay:

DEF click TouchSensor {enabled TRUE }

DEF timer TimeSensor {

cycleInterval 5

loop FALSE

enabled FALSE

}

DEF timer1 TimeSensor {

cycleInterval 5

loop FALSE

enabled FALSE

}

DEF quay1 OrientationInterpolator {

key [0 0.5 1]

keyValue [0 1 0 0,0 1 0 3.14,0 1 0 6.28]

}

DEF quay2 OrientationInterpolator {

key [0 0.5 1]

keyValue [ 0 1 0 0,0 1 0 -3.14,0 1 0 -6.28]

}

DEF ma Script {

eventIn SFBool turnOn

eventOut SFBool On

eventOut SFBool Off

field SFBool turn FALSE

url "javascript:

function turnOn(val){

if(val){

turn = !turn;

if(turn)

{

On= TRUE;

Off=FALSE;

}

Else

{

On=FALSE;

Off=TRUE;

}

}

}

"

}

ROUTE click.isActive TO ma.turnOn

ROUTE ma.On TO timer.enabled

ROUTE ma.Off TO timer1.enabled

ROUTE click.touchTime TO timer.startTime

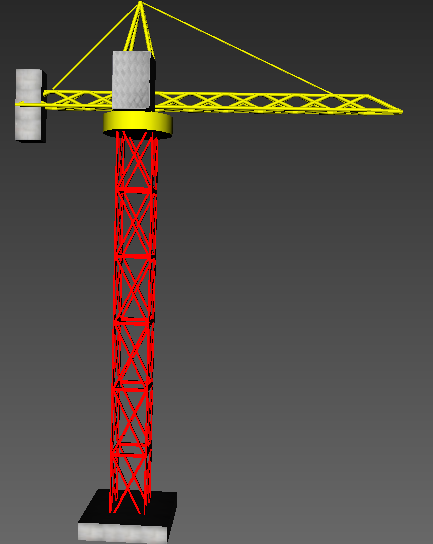
ROUTE click.touchTime TO timer1.startTime

ROUTE timer.fraction\_changed TO quay1.set\_fraction

ROUTE timer1.fraction\_changed TO quay2.set\_fraction

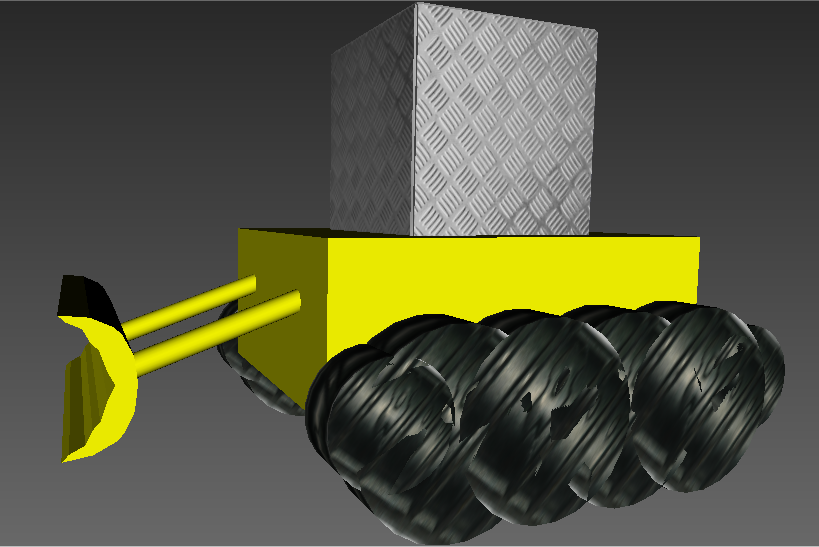
ROUTE quay1.value\_changed TO trucquay.rotation

ROUTE quay2.value\_changed TO trucquay.rotation



## Máy ủi

* Được xây dựng từ các đối tượng khác nhau: lưỡi ủi, bánh xích, hộp và hình hộp.
* Lưỡi ủi được xây dựng từ 2 hình trụ (Cylinder) và 1 hình tạo bởi Extrusion. Lưỡi ủi có thể di chuyển lên xuống khi click chuột bởi nút nội xuy PositionInterpolator và OrientationInterpolator và nhúng mã(Script).
* Bánh xính tạo bởi các hình trụ trồng nhau và quay liên tục bởi nút nội xuy OrientationInterpolator
* Hộp được tạo bởi các tấm hình hộp.



* Máy ủi có thể dịch chuyển khi click chuột bởi nút nội xuy PositionInterpolator và OrientationInterpolator và nhúng mã(Script).
* Code phần dịch chuyển máy ủi:

DEF mayui Transform {

children [Inline {url "mayui.wrl"}]

}

DEF click TouchSensor {enabled TRUE }

DEF timer TimeSensor {

cycleInterval 5

loop FALSE

enabled FALSE

}

DEF timer1 TimeSensor {

cycleInterval 5

loop FALSE

enabled FALSE

}

DEF timer2 TimeSensor {

cycleInterval 5

loop FALSE

enabled FALSE

}

DEF timer3 TimeSensor {

cycleInterval 5

loop FALSE

enabled FALSE

}

DEF tien PositionInterpolator {

key [0 0.5 1]

keyValue [0 0 0,-48 0 0,-48 0 30]

}

DEF quay2 OrientationInterpolator {

key [0.4 0.45 0.5]

keyValue [0 1 0 0,0 1 0 0.785, 0 1 0 1.57]

}

DEF quay1 OrientationInterpolator {

key [0.35 0.4 0.5]

keyValue [0 1 0 1.57,0 1 0 2.355, 0 1 0 3.14]

}

DEF lui PositionInterpolator {

key [0 0.5 1]

keyValue [-48 0 30,-48 0 0 ,0 0 0]

}

DEF ma Script {

eventIn SFBool turnOn

eventOut SFBool On

eventOut SFBool On1

eventOut SFBool On2

eventOut SFBool Off

field SFBool turn FALSE

url "javascript:

function turnOn(val){

if(val){

turn = !turn;

if(turn)

{

On= TRUE;

On1=FALSE;

On2=TRUE;

Off=FALSE;

}

else

{

On=FALSE;

On1=TRUE;

On2=FALSE;

Off=TRUE;

}

}

}

"

}

ROUTE click.isActive TO ma.turnOn

ROUTE ma.On TO timer.enabled

ROUTE ma.Off TO timer1.enabled

ROUTE ma.On1 TO timer2.enabled

ROUTE ma.On2 TO timer3.enabled

ROUTE click.touchTime TO timer.startTime

ROUTE click.touchTime TO timer1.startTime

ROUTE click.touchTime TO timer2.startTime

ROUTE click.touchTime TO timer3.startTime

ROUTE timer.fraction\_changed TO tien.set\_fraction

ROUTE timer1.fraction\_changed TO lui.set\_fraction

ROUTE timer2.fraction\_changed TO quay1.set\_fraction

ROUTE timer3.fraction\_changed TO quay2.set\_fraction

ROUTE tien.value\_changed TO mayui.translation

ROUTE quay1.value\_changed TO mayui.rotation

ROUTE lui.value\_changed TO mayui.translation

ROUTE quay2.value\_changed TO mayui.rotation

## Bao xi măng

* Được xây dựng bằng Extrusion và dán hình.



* Code bao xi măng:

Transform {

children [Shape {appearance Appearance {material Material {}

texture ImageTexture {url "ximang.jpg"}}

geometry Extrusion {

crossSection [1 0, 0.9 0.4, 0.4 0.9, 0 1, -0.4 0.9, -0.9 0.4

-1 0, -0.9 -0.4, -0.4 -0.9, 0 -1 , 0.4 -0.9, 0.9 -0.4, 1 0]

spine [0 0 0,0.2 0 0,0.4 0 0,0.6 0 0,0.8 0 0,1 0 0,3 0 0,3.2 0 0,3.4 0 0,3.6 0 0,3.8 0 0,4 0 0]

scale [0 1,0.1 1,0.3 1,0.35 1,0.4 1,0.5 1,0.5 1,0.4 1,0.35 1,0.3 1,0.1 1,0 1]

beginCap TRUE

endCap TRUE

solid FALSE

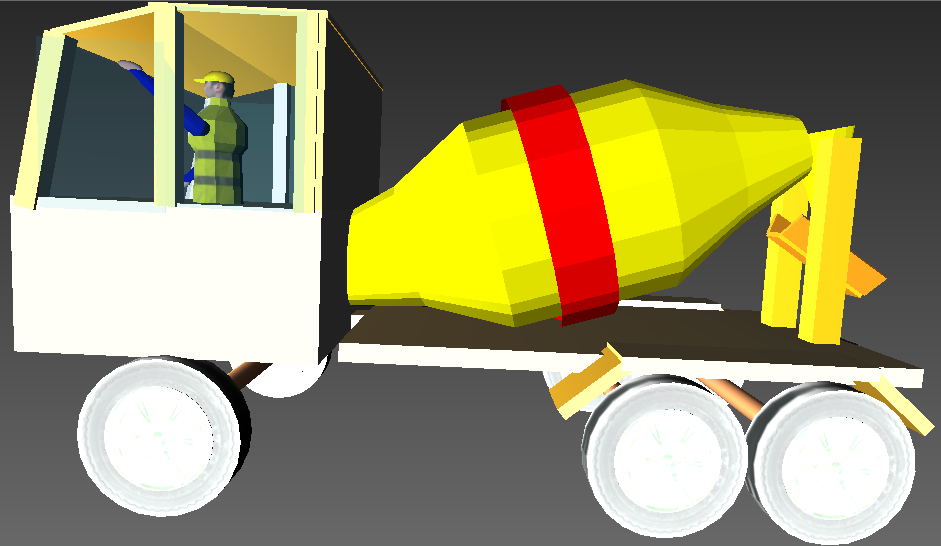
}

}]

}

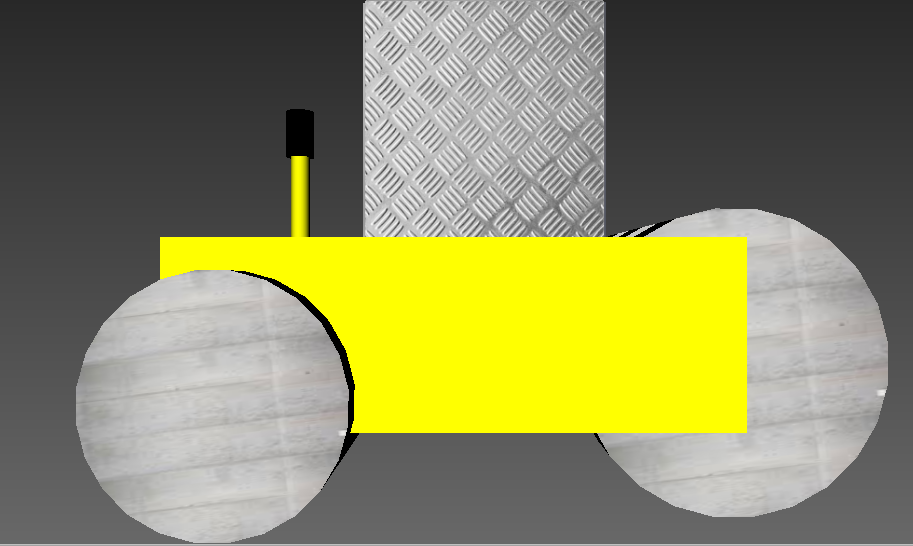
## Xe trộn bê tông

* Gồm 2 phần chính là khung xe và thùng trộn.
* Khung xe được tạo từ các hình hộp(box) và các hình trụ(Cylinder) được ghép nối với nhau bởi phép dịch chuyển (Translation), phép quay (Rotation) và tái sử dụng các đối tượng(DEF).
* Thùng trộn được tạo bởi Extrusion và quay liên tục bởi nút nội xuy OrientationInterpolator



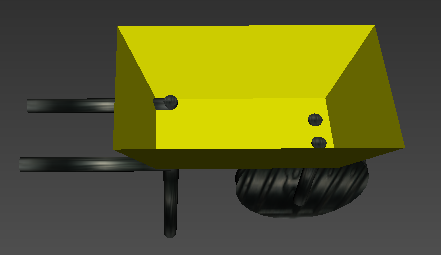
## Xe lu

* Được xây dựng từ các đối tượng khác nhau: bánh, hộp, thân.
* Bánh là các hình trụ quay liên tục bởi nút nội xuy OrientationInterpolator.
* Hộp là đối tượng được gọi đến.
* Thân là hình hộp(box).
* Xe lu có thể di chuyển tiến lùi khi click chuột bởi nút nội xuy PositionInterpolator và nhúng mã(Script).



## Xe rùa

* Gồm các hình tru (Cylinder) và hình được tạo bởi Extrusion.



* Code xe rùa:

Transform {

translation 0 0.3 0

children [Shape {appearance Appearance {material Material {}

texture ImageTexture {url "thep.jpg"}}

geometry Cylinder {radius 0.1 height 2}}

]

}

Transform {

translation 0 0.3 1

children [Shape {appearance Appearance {material Material {}

texture ImageTexture {url "thep.jpg"}}

geometry Cylinder {radius 0.1 height 2}}

]

}

Transform {

translation 2 0.5 0.3

children [Shape {appearance Appearance {material Material {}

texture ImageTexture {url "thep.jpg"}}

geometry Cylinder {radius 0.1 height 1.5}}

]

}

Transform {

translation 2 0.5 0.7

children [Shape {appearance Appearance {material Material {}

texture ImageTexture {url "thep.jpg"}}

geometry Cylinder {radius 0.1 height 1.5}}

]

}

Transform {

translation 2 0 0.5

rotation 1 0 0 1.57

children [Shape {appearance Appearance {material Material {}

texture ImageTexture {url "thep.jpg"}}

geometry Cylinder {radius 1 height 0.2}}

]

}

DEF gau Transform {

translation 1 2.22 0.5

rotation 0 0 1 -1.57

children [

Shape {appearance Appearance {material Material {diffuseColor 1 1 0}}

geometry Extrusion {

crossSection [0 1, 1 1, 0 1, -1 1, -1 0, -1 -1, 0 -1, 1 -1, 1 1 ]

spine [0 0 0, 1 0 0]

scale [1.7 1,1.2 0.6]

beginCap FALSE

endCap TRUE

solid FALSE

}

}

]

}

Transform {

translation -1 1.2 0

rotation 0 0 1 -1.57

children [Shape {appearance Appearance {material Material {}

texture ImageTexture {url "thep.jpg"}}

geometry Cylinder {radius 0.1 height 2}}

]

}

Transform {

translation -1 1.2 1

rotation 0 0 1 -1.57

children [Shape {appearance Appearance {material Material {}

texture ImageTexture {url "thep.jpg"}}

geometry Cylinder {radius 0.1 height 2}}

]

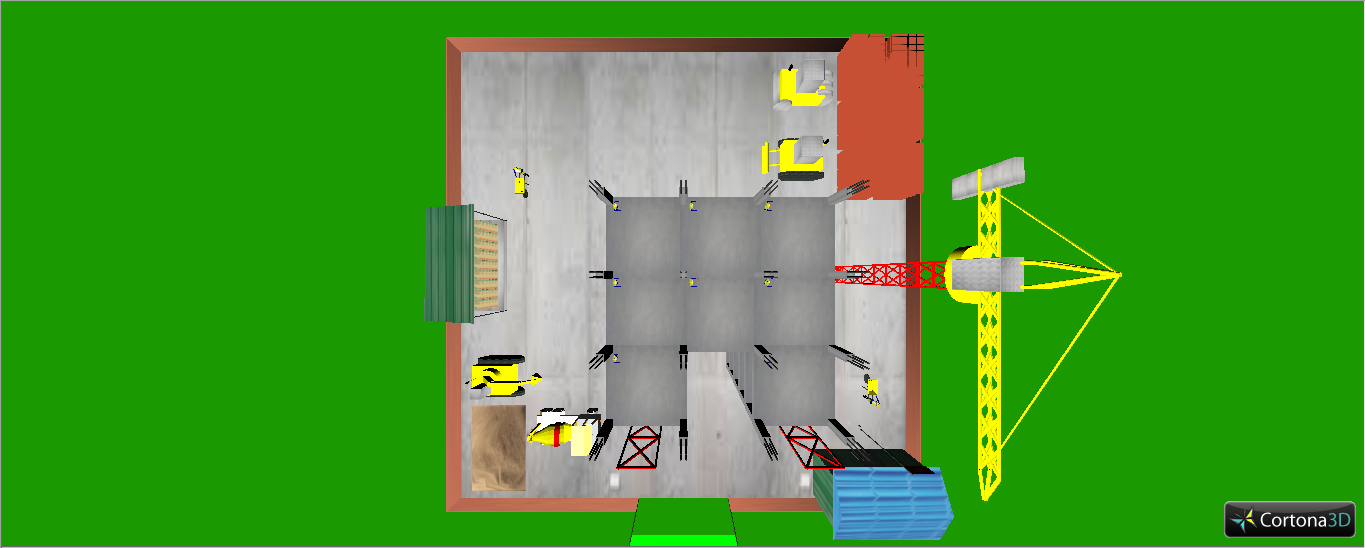
}

## Công trường

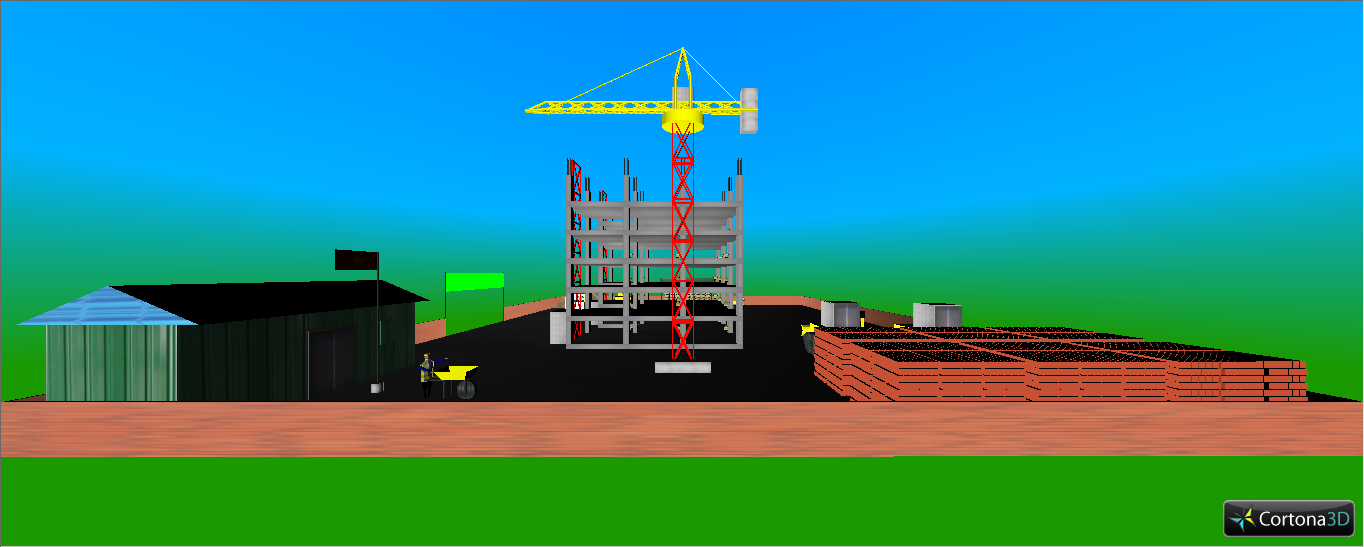
* Bao gồm toàn thể các đối tượng được ghép nối với nhau và đươc thiết lập Camera(Viewpoint) nhìn từ các hướng.
* Công trường nhìn từ phía trước:



* Công trường nhìn từ phía trên xuống:



* Công trường nhìn từ phía bên phải:



* Công trường nhìn từ phía sau:



* Công trường nhìn từ phía bên trái:

