# 虚拟现实大作业开发文档

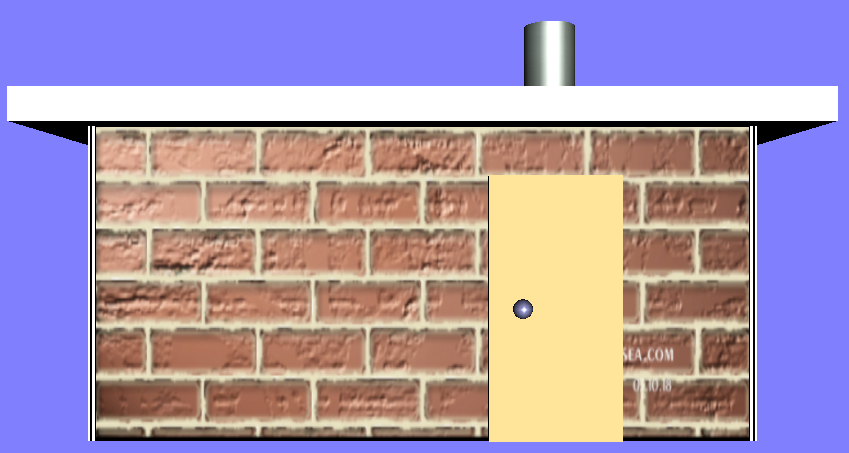
**刘莹 1403121773**

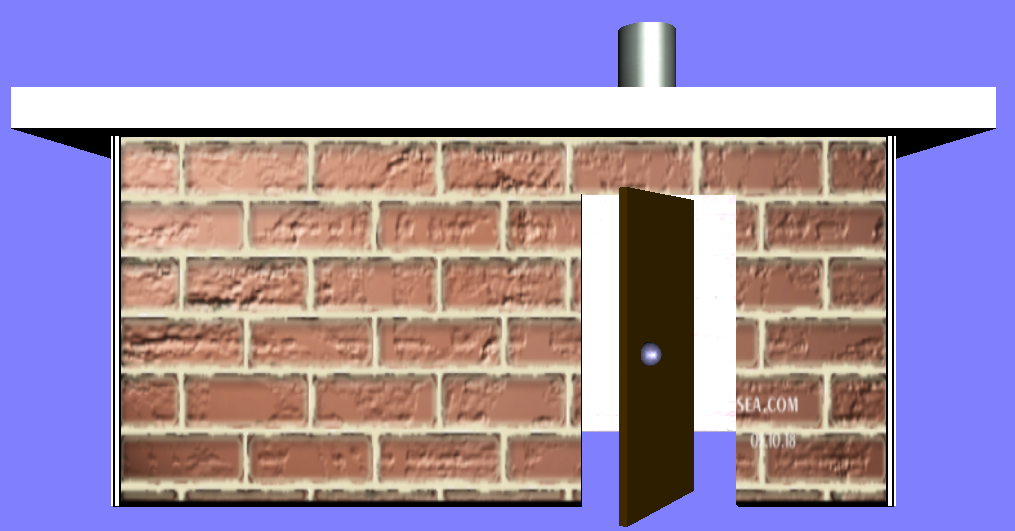
## 功能概述

在VrmlPad中编写程序，构造了一个虚拟世界。主要场景是一个房子，门可以转动。房子内部包括一个时钟和一个电视机。房子后面是很多座风车和两排树。房子内的时钟可以运转，电视机循环播放视频，风车可以转动，场景中还包括背景音乐。

## 二、详细说明

### 1、房子





代码如下：

#VRML V2.0 utf8

Background {

skyColor 0.5 0.5 1

}

DEF house Transform {

children [

DEF walls Group {

children [

DEF box1 Shape { #backside of the house

appearance Appearance { #房子后侧墙面

material Material { #一个长方体

diffuseColor 0.3 0.3 0.5

ambientIntensity 0.3

shininess 0.1

specularColor 0.7 0.7 0.8

}

texture ImageTexture { #纹理贴图

url "my texture/wall.jpg"

}

}

geometry Box {

size 10 5 0.1

}

}

DEF box2 Transform { #leftside of the house

translation -5 0 2.45 #房子左侧墙面

rotation 0 1 0 1.5708

scale 0.5 1 1

children [

USE box1

]

}

Transform { #frontside without the door

translation -5 -2.5 5 #房子正面墙面

rotation 1 0 0 -1.5708

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.3 0.5

ambientIntensity 0.2

shininess 0.1

specularColor 0 0 0

}

texture ImageTexture {

url "my texture/wall.jpg"

repeatS TRUE

repeatT TRUE

}

}

geometry Extrusion { #Extrusion挤出造型

crossSection [ #形状为后侧墙面除去门

0 0,6 0,6 4,8 4,8 0,10 0,10 5,0 5,0 0

]

spine [

0 0 0,

0 0.1 0

]

}

}

]

}

Transform { #rightside of the house

translation 10 0 0 #右侧墙面是左侧墙面的平移

children [USE box2 ]

}

DEF roof Transform { #roof of the house

translation 0 2.5 2.5 #屋顶是一个长方体

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.2 0.2 0.8

ambientIntensity 0.1

shininess 0.15

specularColor 0.8 0.8 0.8

}

texture ImageTexture {

url "my texture/wall6.jpg"

}

}

geometry Box {

size 12 0.5 6

}

}

]

}

DEF chimney Transform { #chimney of the house

translation 2.5 3.75 1.25 #烟囱是一个圆柱体

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.2 0.2 0.8

ambientIntensity 0.1

shininess 0.15

specularColor 0.8 0.8 0.8

}

texture ImageTexture {

url "my texture/wall6.jpg"

}

}

geometry Cylinder {

radius 0.5

height 2

}

}

]

}

]

}

Group { #door of the house

children [ #房门是长方体加球形门把手

DEF door Transform {

translation 2 -0.5 5

children [

Group{

children [

Transform {

translation 0 0 0

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.2 0.0

ambientIntensity 0.4

shininess 0.2

specularColor 0.7 0.7 0.6

transparency 0.0

}

}

geometry Box {

size 2 4 0.1

}

}

]

}

Transform { #the doornob

translation -0.5 0 0.1

children [ #球形门把手

Shape {

appearance Appearance {

material Material {

diffuseColor 0.5 0.5 0.7

ambientIntensity 0.4

shininess 0.2

specularColor 0.8 0.8 0.9

transparency 0

}

}

geometry Sphere {

radius 0.15

}

}

]

}

]

}

DEF door\_cs CylinderSensor { #开门动作由圆柱检测器控制

autoOffset TRUE

diskAngle 0.262

enabled TRUE

autoOffset TRUE

maxAngle 1.5708 #最大开门角度是90°

minAngle 0.0

offset 1.576

}

]

}

]

}

]

}

ROUTE door\_cs.rotation\_changed TO door.set\_rotation

### 2、电视



代码如下：

#VRML V2.0 utf8

Transform { #biankuang

translation 0 0 0 #电视边框

children [ #一个窄长方体

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.3 0.3

}

}

geometry Box {

size 5.1 3.4 0.2

}

}

]

}

Transform { #pingmu

translation 0 0 0.1 #电视屏幕

children [ #一个窄长方体

Shape {

appearance Appearance {

texture DEF film MovieTexture { #电影纹理，控制播放视频

url "my file/xqdz1.MPG"

loop TRUE

}

}

geometry Box {

size 4.5 3 0.01

}

}

]

}

Transform { #zhijia

translation 0 -2 0 #电视支架

children [ #一个圆柱体

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.3 0.3

}

}

geometry Cylinder {

radius 0.1

height 0.6

}

}

]

}

Transform { #dizuo

translation 0 -2.25 0 #电视底座

rotation 1 0 0 1.5708 #一个扁长方体

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.3 0.3

}

}

geometry Box {

size 3 2 0.2

}

}

]

}

Transform { #guizi

translation 0 -3.25 0 #用来放电视的柜子

children [ #一个长方体

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.3 0.3

}

texture ImageTexture {

url "my texture/wood.jpg"

}

}

geometry Box {

size 5.5 2 3.5

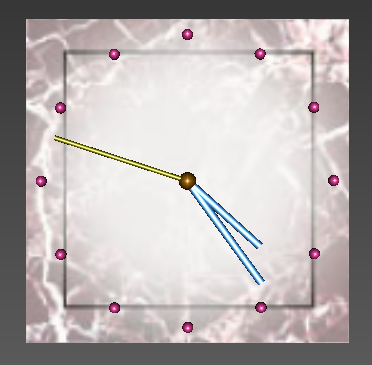
}

}

]

}

### 3、时钟



代码如下：

#VRML V2.0 utf8

DEF clock Group {

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.8 0.8 0.8 }

texture ImageTexture { #给表盘纹理贴图

url "my texture/clock.jpg"

}

}

geometry Box { size 4 4 0.2 } }

DEF time1 TimeSensor {

cycleInterval 1

loop TRUE

enabled TRUE }

DEF an Transform {

translation 0 0 0.15

children [

DEF ts0 TouchSensor {}

Shape {

appearance Appearance {

material Material {

diffuseColor 0.5 0.3 0

ambientIntensity 0.4

specularColor 0.7 0.7 0.6

shininess 0.2 } }

geometry Sphere { radius 0.11 }

} ] }

Transform {

translation 0 0 0.15

children [

DEF p1 Transform {

translation 0 0.9 0.05

children [

DEF ts1 CylinderSensor {} #秒针

Shape {

appearance Appearance {

material Material {

diffuseColor 0.5 0.5 0

ambientIntensity 0.4

specularColor 0.8 0.8 0.9

shininess 0.2 } }

geometry Cylinder {

height 1.62

radius 0.03 }

} ] }

DEF p2 Transform { #分针

translation 0 0.8 0

children [

DEF ts2 CylinderSensor {}

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.6 0.9

ambientIntensity 0.4

specularColor 0.8 0.8 0.9

shininess 0.1 } }

geometry Cylinder {

height 1.5

radius 0.05 }

} ] }

DEF p3 Transform { #时针

translation 0 0.6 0

children [

DEF ts3 CylinderSensor {}

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.6 0.9

ambientIntensity 0.4

specularColor 0.8 0.8 0.9

shininess 0.1 } }

geometry Cylinder {

height 1.2

radius 0.05 }

} ] }

DEF bkd Transform { #整点位置

translation 0 1.8 0

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.8 0.2 0.5

ambientIntensity 0.4

specularColor 0.8 0.8 0.9

shininess 0.2 } }

geometry Sphere {

radius 0.07 }

} ] }

Transform {

rotation 0 0 1 0.524

children [

USE bkd ]}

Transform {

rotation 0 0 1 1.048

children [

USE bkd ]}

Transform {

rotation 0 0 1 1.572

children [

USE bkd ]}

Transform {

rotation 0 0 1 2.096

children [

USE bkd ]}

Transform {

rotation 0 0 1 2.620

children [

USE bkd ]}

Transform {

rotation 0 0 1 3.144

children [

USE bkd ]}

Transform {

rotation 0 0 1 3.668

children [

USE bkd ]}

Transform {

rotation 0 0 1 4.192

children [

USE bkd ]}

Transform {

rotation 0 0 1 4.716

children [

USE bkd ]}

Transform {

rotation 0 0 1 5.240

children [

USE bkd ]}

Transform {

rotation 0 0 1 5.764

children [

USE bkd ]}

]}

DEF Controller Script {

eventIn SFTime clicked

eventOut SFBool enabledt

eventOut SFBool setenabled

field SFInt32 i 1

field SFInt32 j 1

field SFInt32 k 1

field SFInt32 on 0

field SFFloat sz -0.10472

field SFFloat mz -0.10472

field SFFloat hz -0.10472

eventOut SFTime miao

eventOut SFTime fen

eventOut SFTime xshi

eventIn SFTime sec\_xz

eventIn SFRotation xuanzhuan\_sec

eventIn SFRotation xuanzhuan\_min

eventIn SFRotation xuanzhuan\_hou

field SFNode sec USE p1

field SFNode min USE p2

field SFNode hou USE p3

url "vrmlscript:

function initialize(){

print('ťԵʱ룡');

enabledt=TRUE;

setenabled=TRUE;

mydate=new Date();

year=mydate.getFullYear();

mont=mydate.getMonth();

date=mydate.getDate();

hour=mydate.getHours();

minu=mydate.getMinutes();

seco=mydate.getSeconds();

deff1=seco;

deff2=minu\*60+seco;

print('ʱ䣺='+year+''+mont+''+date+''+hour+''+minu+''+seco+'');

if (deff2>720){ //ʱת1720=3600\*12/60

deff2=deff2-720\*Math.floor(deff2/720); }

hour1=hour+minu/60;

loop=2\*3.14159\*hour1/12/0.10472;

while(k<=loop){

xuanzhuan\_hou();}

minu1=minu+seco/60;

loop=2\*3.14159\*minu/60/0.10472;

while(j<=loop){

xuanzhuan\_min(); }

loop=2\*3.14159\*seco/60/0.10472+1;

while(i<=loop){

xuanzhuan\_sec(); }

setenabled=FALSE;

}

function clicked(value){

enabledt=!enabledt;

setenabled=!setenabled; }

function sec\_xz(a){

a=a+hour\*3600+minu\*60+seco;

if (on==0){

on=1;

miao=a;

fen=a;

xshi=a; }

if (a-miao>=1){

hd=1.57+sz\*i;

y=0.9\*Math.sin(hd);

x=0.9\*Math.cos(hd);

z=0.05;

with (sec){

rotation[0]=0;

rotation[1]=0;

rotation[2]=1;

rotation[3]=sz\*i;

translation[0]=x;

translation[1]=y;

translation[2]=z; }

i=i+1;

miao=a; }

if (a+deff1-fen>=60){

hd=1.57+sz\*j;

y=0.8\*Math.sin(hd);

x=0.8\*Math.cos(hd);

z=0;

min.rotation[0]=0;

min.rotation[1]=0;

min.rotation[2]=1;

min.rotation[3]=sz\*j;

min.translation[0]=x;

min.translation[1]=y;

min.translation[2]=z;

j=j+1;

fen=a;

deff1=0; }

if (a+deff2-xshi>=720){ //ʱת1720=3600\*12/60

hd=1.57+sz\*k;

y=0.6\*Math.sin(hd);

x=0.6\*Math.cos(hd);

z=0;

hou.rotation[0]=0;

hou.rotation[1]=0;

hou.rotation[2]=1;

hou.rotation[3]=sz\*k;

hou.translation[0]=x;

hou.translation[1]=y;

hou.translation[2]=z;

k=k+1;

xshi=a;

deff2=0; } }

function xuanzhuan\_sec(){

if (setenabled){

hd=1.57+sz\*i;

y=0.9\*Math.sin(hd);

x=0.9\*Math.cos(hd);

z=0.05;

sec.rotation[0]=0;

sec.rotation[1]=0;

sec.rotation[2]=1;

sec.rotation[3]=sz\*i;

sec.translation[0]=x;

sec.translation[1]=y;

sec.translation[2]=z;

i=i+1; } }

function xuanzhuan\_min(){

if (setenabled){

hd=1.57+sz\*j;

y=0.8\*Math.sin(hd);

x=0.8\*Math.cos(hd);

z=0;

min.rotation[0]=0;

min.rotation[1]=0;

min.rotation[2]=1;

min.rotation[3]=sz\*j;

min.translation[0]=x;

min.translation[1]=y;

min.translation[2]=z;

j=j+1; } }

function xuanzhuan\_hou(){

if (setenabled){

hd=1.57+sz\*k;

y=0.6\*Math.sin(hd);

x=0.6\*Math.cos(hd);

z=0;

hou.rotation[0]=0;

hou.rotation[1]=0;

hou.rotation[2]=1;

hou.rotation[3]=sz\*k;

hou.translation[0]=x;

hou.translation[1]=y;

hou.translation[2]=z;

k=k+1; } }

" }

]

}

ROUTE time1.time TO Controller.sec\_xz

ROUTE ts0.touchTime TO Controller.clicked

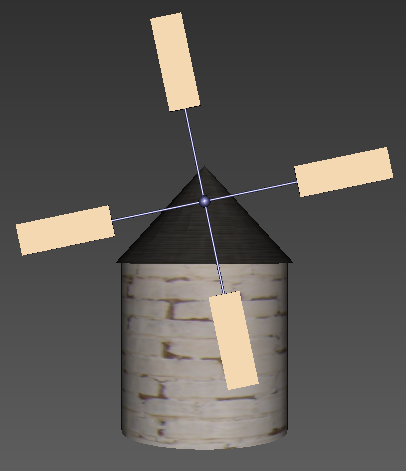
ROUTE Controller.enabledt TO time1.enabled

ROUTE ts1.rotation\_changed TO Controller.xuanzhuan\_sec

ROUTE ts2.rotation\_changed TO Controller.xuanzhuan\_min

ROUTE ts3.rotation\_changed TO Controller.xuanzhuan\_hou

### 4、风车



代码如下：

#VRML V2.0 utf8

Group { #该组包括两个互相垂直的扇叶和中间部分金属球

children [

DEF bars Transform { #定义一个名为bars的

translation 0 5 3.5 #内容为互相垂直的扇叶

children [

DEF bar1 Transform{ #定义bar1

translation 0 0 0 #bar1包括一个杆和两片扇叶

children [

Transform {

translation 0 0 0

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.3 0.5

ambientIntensity 0.3

shininess 0.1

specularColor 0.7 0.7 0.8

}

}

geometry Cylinder { #圆柱形杆

radius 0.04

height 12

}

}

]

}

DEF a1 Transform { #长方体形扇叶

translation 0 4.5 0

rotation 1 0 0 1.5708

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.6 0.9

}

texture ImageTexture {

url "my texture/wall4.jpg"

}

}

geometry Box {

size 1 0.1 3

}

}

]

}

DEF a2 Transform {

translation 0 -9 0

children [

USE a1

]

}

]

}

DEF bar2 Transform { #bar2是将bar1沿z轴旋转90°

rotation 0 0 1 1.5708

children [

USE bar1

]

}

Transform { #两个bar交点处的金属球

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.3 0.5

ambientIntensity 0.3

shininess 0.1

specularColor 0.7 0.7 0.8

}

}

geometry Sphere {

radius 0.2

}

}

]

}

]

}

DEF Time TimeSensor { #时间传感器

cycleInterval 6.0 #周期为6

loop TRUE #循环转动

}

DEF fengche OrientationInterpolator { #朝向插补器控制风车转动

key [

0.0,0.2,0.4,0.6,0.8,1.0

]

keyValue [ #绕Z轴转动

0.0 0.0 1.0 0.0 #一个周期转动一周

0.0 0.0 1.0 1.256

0.0 0.0 1.0 2.512

0.0 0.0 1.0 3.768

0.0 0.0 1.0 5.024

0.0 0.0 1.0 6.280

]

}

]

}

ROUTE Time.fraction\_changed TO fengche.set\_fraction

ROUTE fengche.value\_changed TO bars.set\_rotation

Transform { #风车主体

children [ #一个圆柱体

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.6 0.9

}

texture ImageTexture {

url "my texture/wall2.jpg"

}

}

geometry Cylinder {

radius 3

height 6

}

}

]

}

Transform { #风车主体的顶部

translation 0 4.75 0 #一个圆锥体

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.6 0.9

}

texture ImageTexture {

url "my texture/wall3.jpg"

}

}

geometry Cone {

bottomRadius 3.2

height 3.5

}

}

]

}

Transform { #用来连接扇叶和主体顶部的杆

translation 0 5 2.5 #圆柱形金属杆

rotation 1 0 0 1.5708

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.3 0.3 0.5

ambientIntensity 0.3

shininess 0.1

specularColor 0.7 0.7 0.8

}

}

geometry Cylinder {

radius 0.1

height 2.3

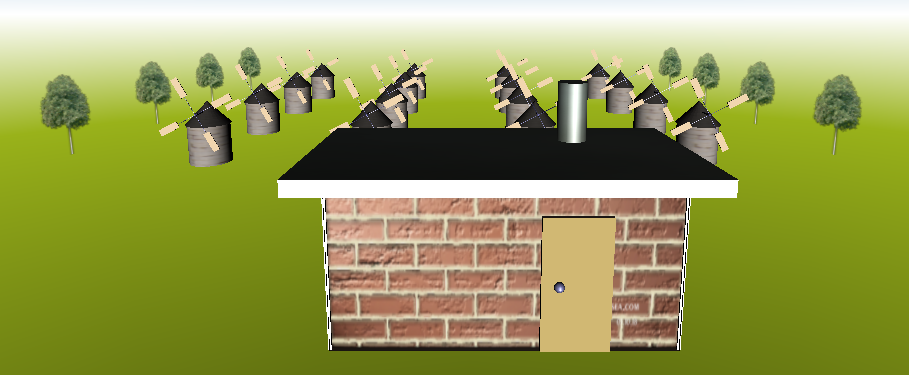
}

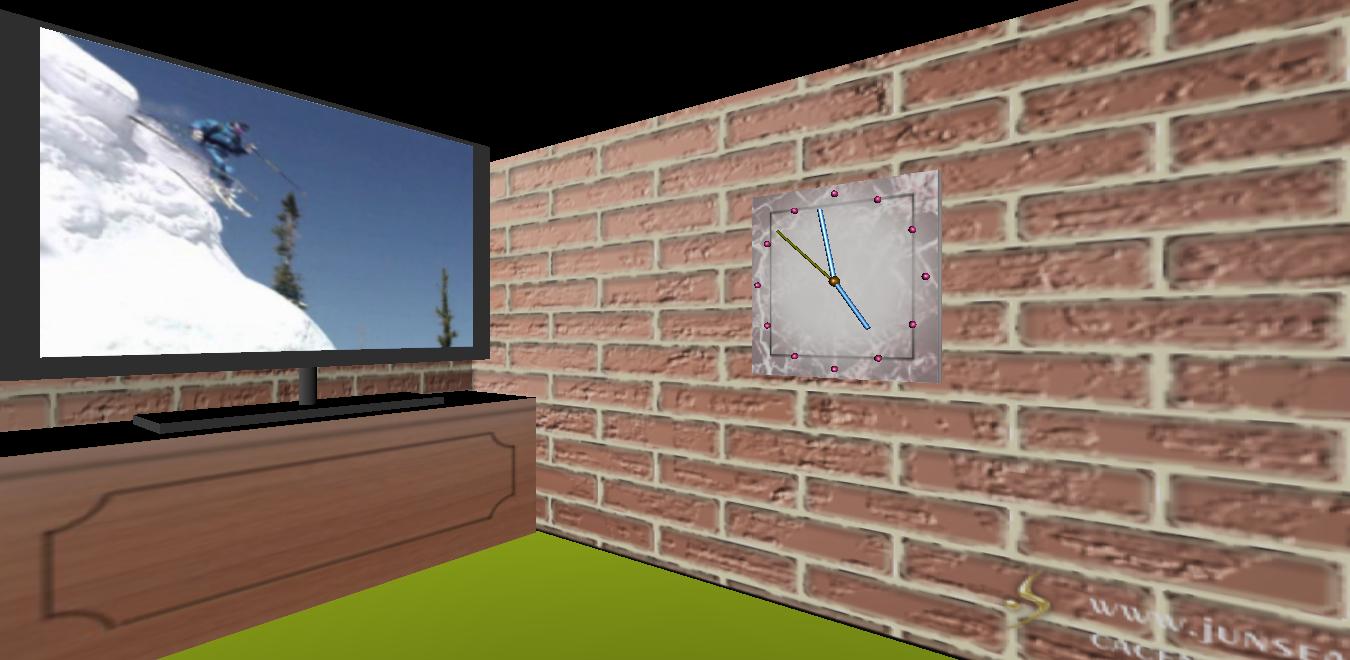
}

]

}

## 5、vrml主程序





代码如下：

#VRML V2.0 utf8

DEF outside Viewpoint{ #two viewpoints

position -5 35 110 #设置第一个视点，可以看到整体场景

orientation -1 0 0 0.15

fieldOfView 1.1717

description "outside"

}

DEF inside Viewpoint { #设置第二个视点，看到室内场景

position 12 0 45

orientation 0 1 0 0.65

fieldOfView 1

description "inside"

}

Background { #skycolor and groundcolor

skyAngle [1.309 1.571] #设置背景颜色

skyColor [

0.0 0.0 0.8

0.2 0.5 0.7

1.0 1.0 1.0

]

groundAngle [1.396 1.571]

groundColor [

0.0 0.0 0.0

0.6 0.7 0.1

1.0 1.0 1.0

]

}

Transform { #inline a house

translation 0 0 20 #内联房子程序

scale 5 5 5 #调整坐标位置和大小

children [

Inline {

url "house.wrl"

}

]

}

#inline a clock

Transform { #内联时钟程序

translation 2 2 20.8

scale 2 2 3

children [

Inline {

url "clock.wrl"

}

]

}

Transform { #inline a television

translation -20 5 32.5 #内联电视程序

rotation 0 1 0 1.5708 #调整方向

scale 4 4 2

children [

Inline {

url "TV.wrl"

}

]

}

DEF fcs Group { #a group of windmills

children [ #内联风车程序，并复制出一列风车

DEF fc Transform {

translation -35 -4 -40

scale 2 2 2

children [

Inline {

url "fengche.wrl"

}

]

}

Transform {

translation 0 0 -40

children [

USE fc

]

}

Transform {

translation 0 0 -80

children [

USE fc

]

}

Transform {

translation 0 0 -120

children [

USE fc

]

}

]

}

Transform { #移动复制一列风车，形成方阵

translation 50 0 0

children [

USE fcs

]

}

Transform {

translation 100 0 0

children [

USE fcs

]

}

Transform {

translation -50 0 0

children [

USE fcs

]

}

DEF trees Group { #a group of trees

children [ #一组树

DEF tree Transform {

translation 100 -5 -30

scale 3 3 3

children [

Billboard {

children [

Shape {

appearance Appearance {

material Material {

diffuseColor 0.8 0.8 0.8

ambientIntensity 0.2

}

texture ImageTexture {

url "my texture/tree.PNG"

}

}

geometry IndexedFaceSet { #形状为索引面

coord Coordinate {

point [

-2 0 0.015

2 0 0.015

-2 8 0.015

2 8 0.015

]

}

coordIndex [0,1,3,2,-1]

texCoord TextureCoordinate {

point [

0.005 0.005,

0.995 0.005,

0.005 0.995,

0.995 0.995

]

}

}

}

]

axisOfRotation 0 1 0

}

]

}

Transform { #复制，形成一列树

translation 0 0 -40

children [

USE tree

]

}

Transform {

translation 0 0 -80

children [

USE tree

]

}

Transform {

translation 0 0 -120

children [

USE tree

]

}

]

}

Transform { #将上面的一列树复制移动

translation -220 0 0

children [

USE trees

]

}

Sound { #background music

source AudioClip { #添加背景音乐

url "my file/Bandari liulihupan.mid" #班得瑞的《琉璃湖畔》

description "sound"

loop TRUE

pitch 1.0

}

direction 0 0 1

intensity 1

location 0 0 0

maxBack 500

maxFront 500

minBack 0

minFront 0

spatialize TRUE

}

## 三、开发过程概述

在课堂上听老师分析实例，课下参考书籍了解原理，然后在网上搜集vrml程序，熟悉编程语言。然后着手编写简单事物模型，再在静态模型基础上添加贴图和动画。然后将各个事物模型添加到一个场景中，不断完善，构成了最终的虚拟场景。

变成过程中遇到过很多问题，反复琢磨之后找到了原因并改正。能掌握一项新的技能，自己还是很满足的！

## 四、运行环境说明

编程环境：WIN7

编程软件：VrmlPad

浏览器：Cortona 3D Viewer