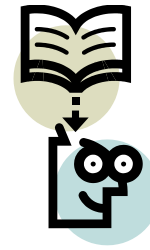
فصل سوم

شبیه سازی انتقال مکعب



۳-۱. مقدمه

هدف این فصل، معرفی جعبه ابزار واقعیت مجازی برای کاربران متلب که می خواهند روابط و سیستم های فیزیکی خود را شکل گرافیکی بدهند و دینامیک سیستم های فیزیکی خود را به صورت انیمیشن، در دنیای مجازی شبیه سازی نمایند.

در این فصل، به نحوه ساخت شکل های گرافیکی با استفاده از نرم افزار VRML و ساخت انیمیشن برای معادلات حالت سیستم فیزیکی، پرداخته خواهد شد.

در بخش ۳-۲ یک مساله دینامیکی ساده انتقال مکعب که تحت تاثیر نیروی خارجی قرار گرفته است در نظر گرفته می شود. معادلات حرکت سیستم با استفاده از قانون دوم نیوتن نوشته می شوند.

در بخش ۳-۳ مراحل ساخت مکعب، تغییر نام، رنگ و اندازه آن مرحله به مرحله آموزش داده می شود.

قرارداد محورهای مختصات x، y، z نرم افزار VRML بطور مختصر در بخش ۳-۴ بیان خواهد شد.

در بخش ۳-۵ ، برنامه متلب برای حل عددی معادلات دیفرانسیل حاکم بر دینامیک سیستم نوشته خواهد شد.

برای ایجاد انیمیشن صحنه مجازی، یک برنامه متلب دیگر نوشته خواهد شد (بخش ۶-۳) که این برنامه از خروجی حل عددی برنامه نوشته شده در قسمت ۵-۳ استفاده می کند و انیمیشن ساخته شده را، با فرمت فایل ویدیویی در مسیر جاری نرم افزار متلب ذخیره می نماید.

۲-۳. مساله انتقال مکعب

مکعب به جرم $m = 1\ (kg)$ ، به ابعاد $0.5 \times 0.5 \times 0.5\ (m^3)$ تحت تاثیر نیروی خارجی در امتداد محور z ، $F_z = -1\ (N)$ ، در نظر گرفته می شود.

با استفاده از قانون دوم نیوتن، معادله دیفرانسیل مرتبه دو خطی، برای جابجایی در جهت محور z ، نوشته می شود. (برای قرارداد سیستم محورهای مختصات به بخش ۴-۳ رجوع کنید).

$$m\ddot{z}(t) = F_z \Rightarrow \ddot{z}(t) = \frac{F_z}{m} \qquad (۳.۱)$$

هدف این مساله، ساخت انیمیشن برای حل این معادله (جابجایی مکعب در امتداد Z) می باشد.

۳-۳. ساخت صحنه مجازی برای انتقال مکعب

برای ساخت صحنه مجازی برای مکعب، مراحل زیر را دنبال کنید.

۱- نرم افزار V-Realm Builder را باز کرده و بر روی File > New کلیک کنید تا دنیای مجازی جدید ایجاد شود.

۲- بر روی دکمه Insert Background کلیک کنید (شکل ۱-۳) تا یک پس زمینه برای دنیای مجازیتان ایجاد شود.

۳- بر روی Insert Box کلیک کنید (شکل ۲-۳) تا یک مکعب ایجاد شود. در پنجره main view شما باید یک مکعب با پس زمینه سبز و آبی ببینید. همچنین در سمت چپ پنجره main view ، عبارات متنی و آیکن های زرد رنگ ایجاد می شوند.همه آنها زیر نود عنوان Transform می باشند.(شکل ۳-۳).

۴- یکبار بر روی عنوان Transform کلیک کنید تا انتخاب شود، بعد از چند لحظه دوباره بروی عنوان Transform انتخاب شده کلیک کرده و نام آن را به Cube تغییر دهید.توصیه می شود در نام گذاری نود Transform از عبارت و نام با معنا برای هر پروژه استفاده کنید.(شکل ۴-۳).

۵- بر روی دکمه Color Mode کلیک کنید.(شکل ۵-۳).

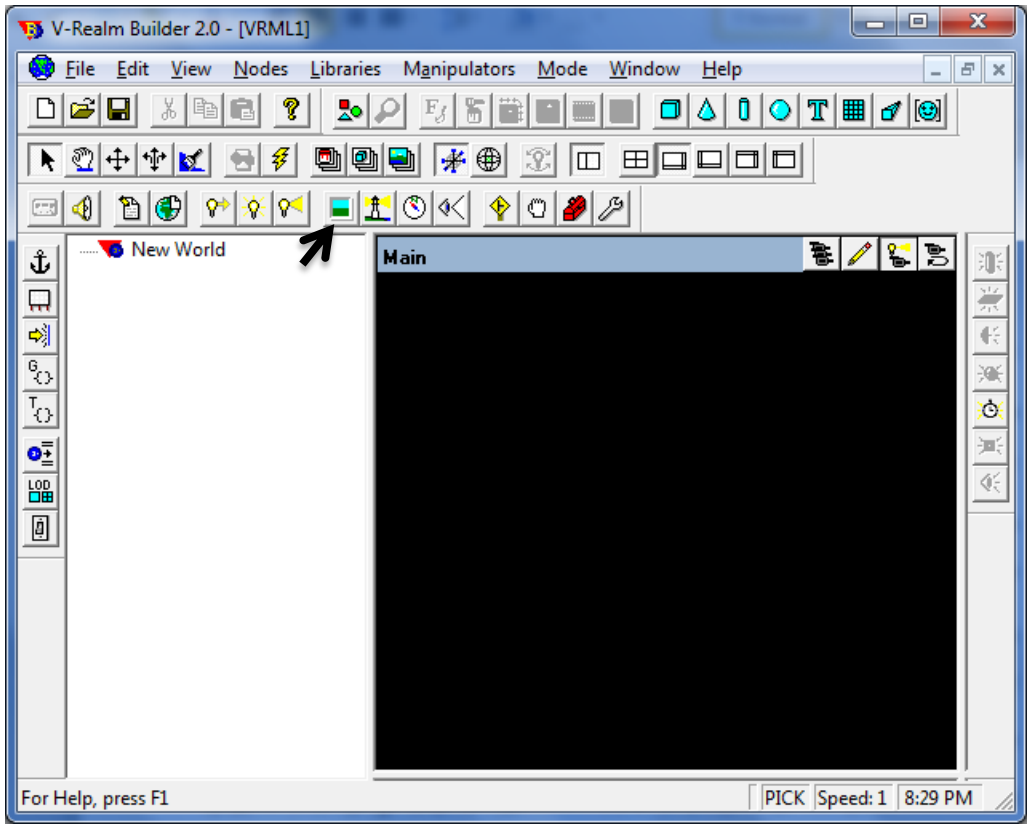
۶- بر روی ناحیه خاکستری Diffusive Color دابل کلیک کنید.(شکل ۶-۳).

۷- بر روی رنگ آبی یا هر رنگ دلخواه کلیک کنید.(شکل ۷-۳). در پایان بر روی دکمه ok کلیک کنید.

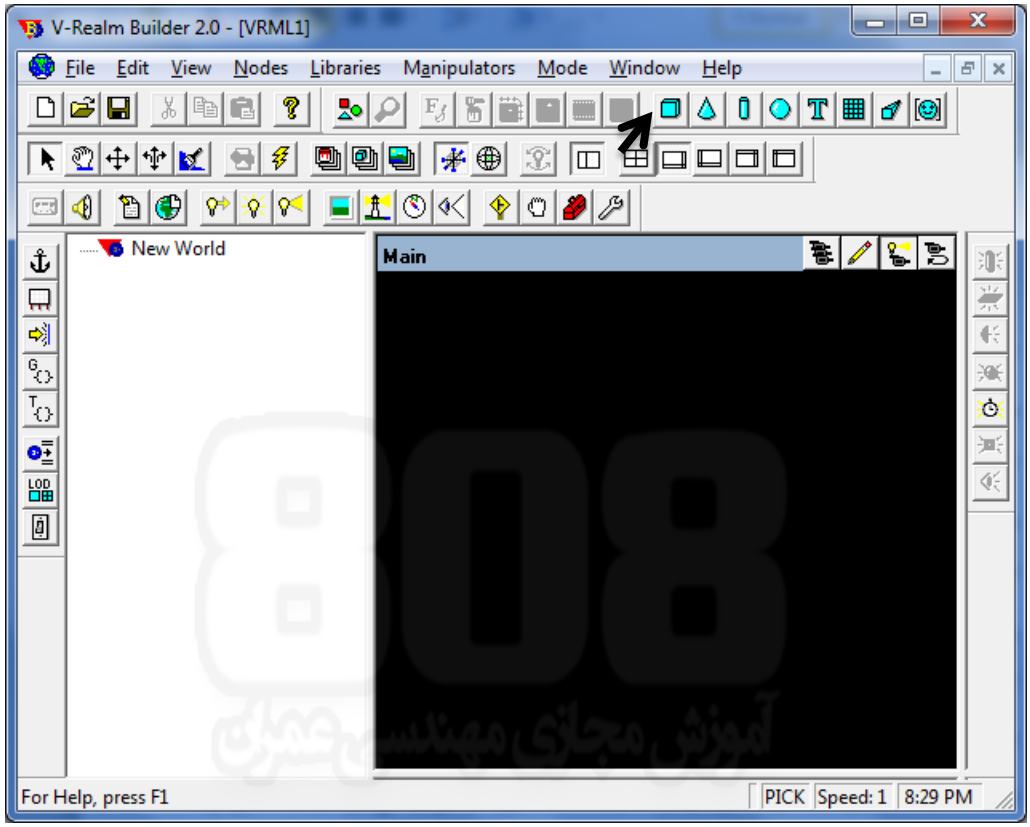
۸- بر روی وجه دلخواه مکعب کلیک کنید، تمامی وجه های شی گرافیکی (مکعب) به رنگ آبی در می آید.(شکل ۸-۳). در پایان بر روی دکمه Close کلیک کنید.

۹- بطور پیشفرض ابعاد مکعب $2 \times 2 \times 2\ (m^3)$ و مرکز آن در $(0,0,0)$ می باشد. برای تغییر ابعاد مکعب ، بروی علامت + در سمت چپ Box کلیک کنید و سپس بر روی Size دابل کلیک کنید .(شکل ۹-۳). جعبه x Axis را تیک زده و مقدار آن را به ۰.۵ تغییر دهید.بطور مشابه بروی جعبه های y Axis و z Axis کلیک کنید و مقدار پیشفرض آن را به ۰.۵ تغییر دهید.در انتها بروی ok کلیک کنید.

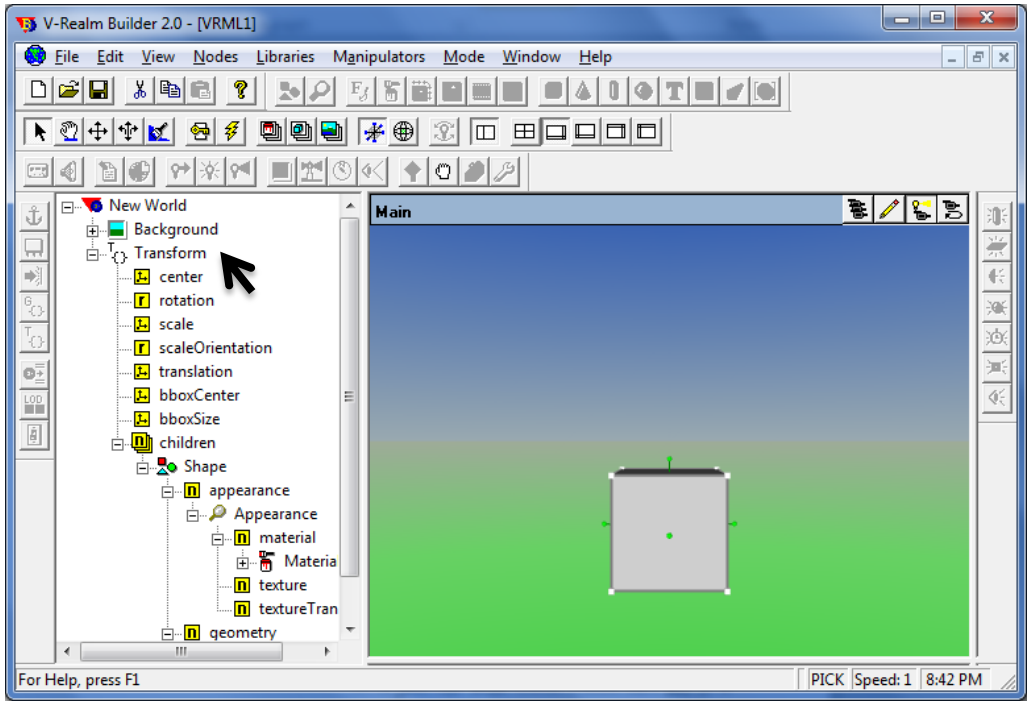
۱۰- بروی File > Save as کلیک کنید و آن را با نام Cube_Virtual در مسیر جاری نرم افزار متلب ذخیره کنید. پسوند فایل واقعیت مجازی wrl. می باشد.



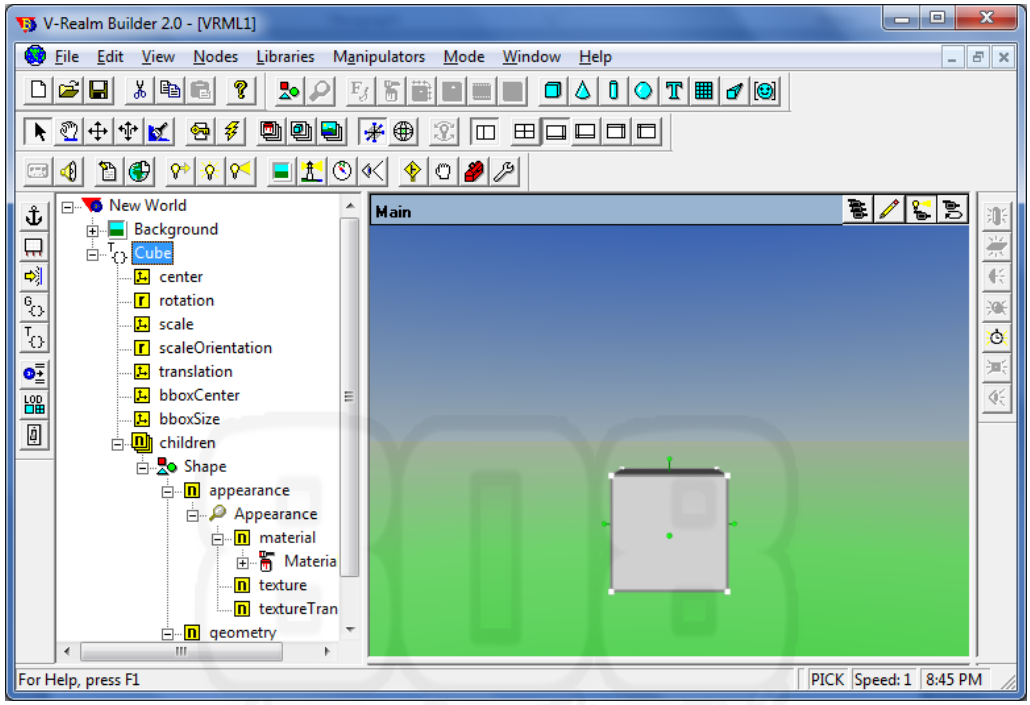
شکل ۱-۳ وارد کردن Background ، (نشان داده شده در شکل)



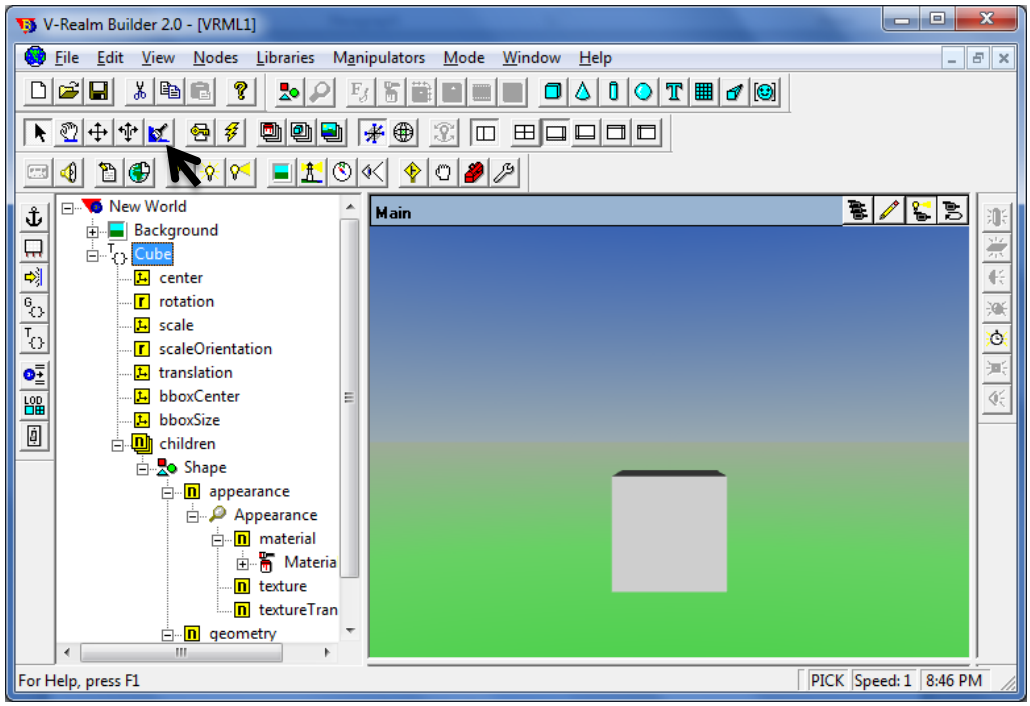
شکل ۲-۳ وارد کردن Box



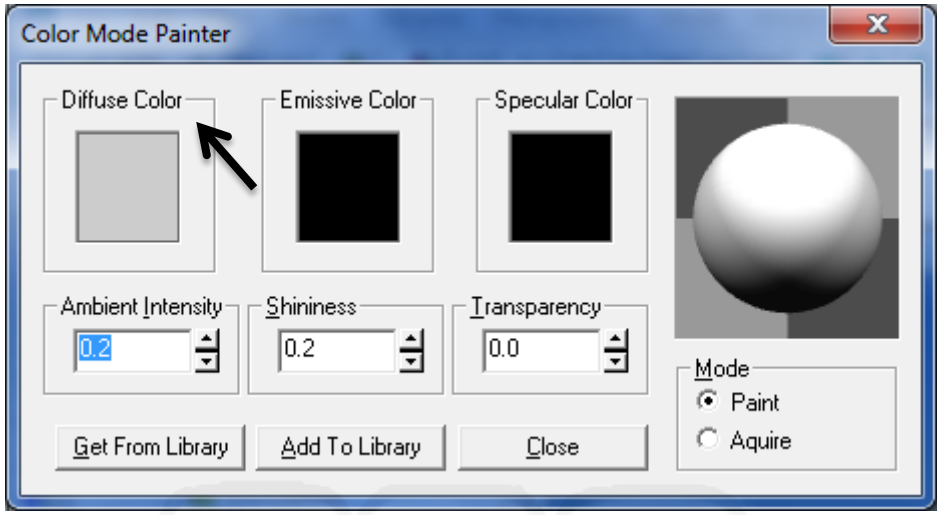
شکل ۳-۳ متن و آیکون های ظاهر شده زیر عنوان Transform



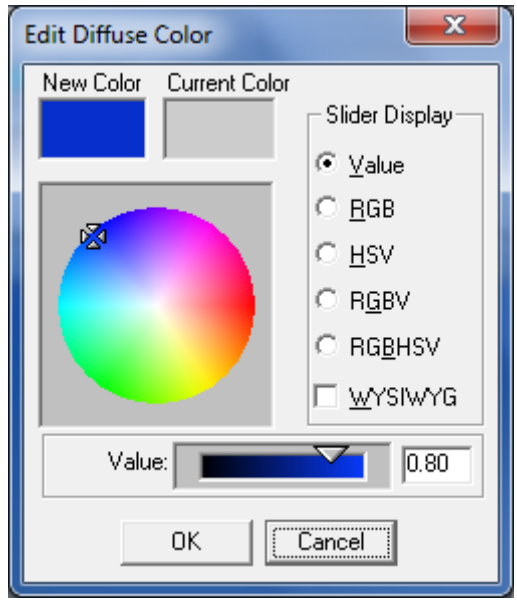
شکل ۴-۳ تغییر نام مکعب از Transform به Cube



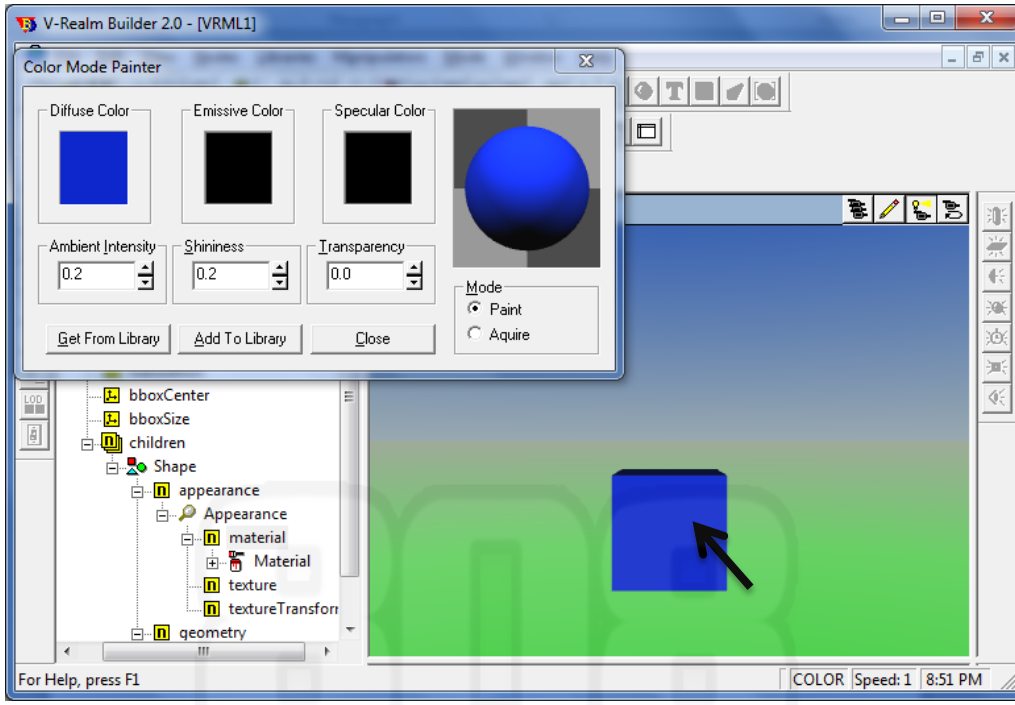
شکل ۵-۳ کلید Color Mode



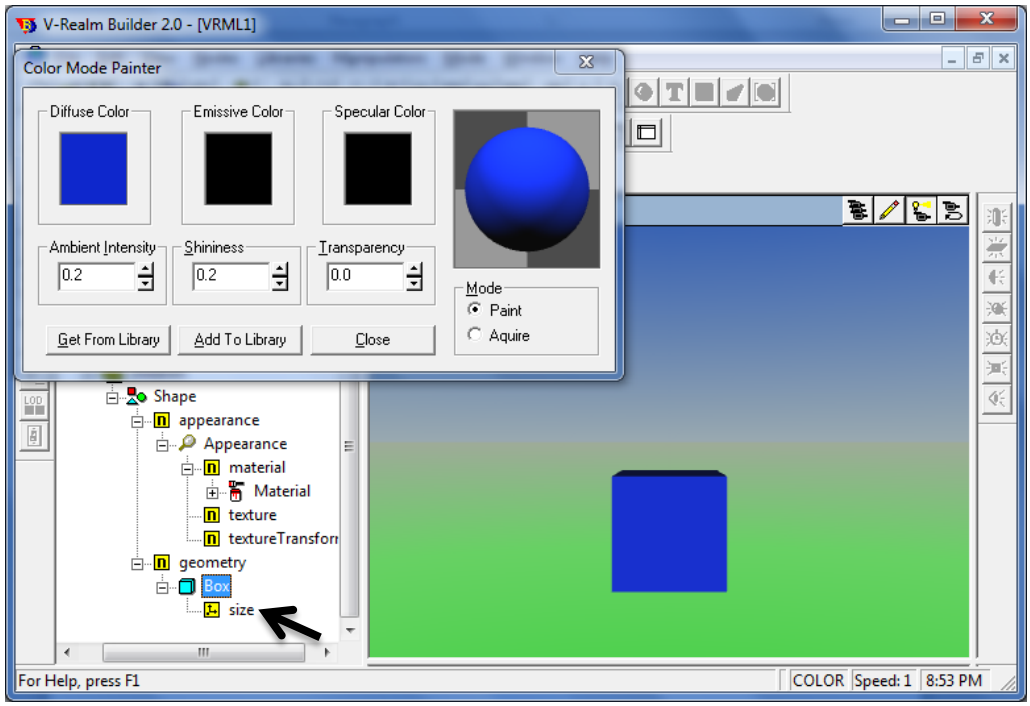
شکل ۶-۳ ناحیه Diffusive Color gray



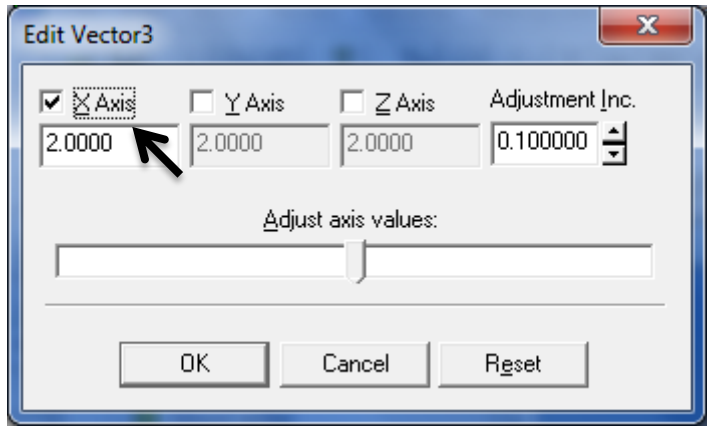
شکل ۷-۳ بروی ناحیه آبی کلیک کنید تا رنگ آبی انتخاب شود



شکل ۸-۳ بروی وجه دلخواه مکعب کلیک کنید



شکل ۹-۳ خصوصیت Size نشان داده شده زیر Box



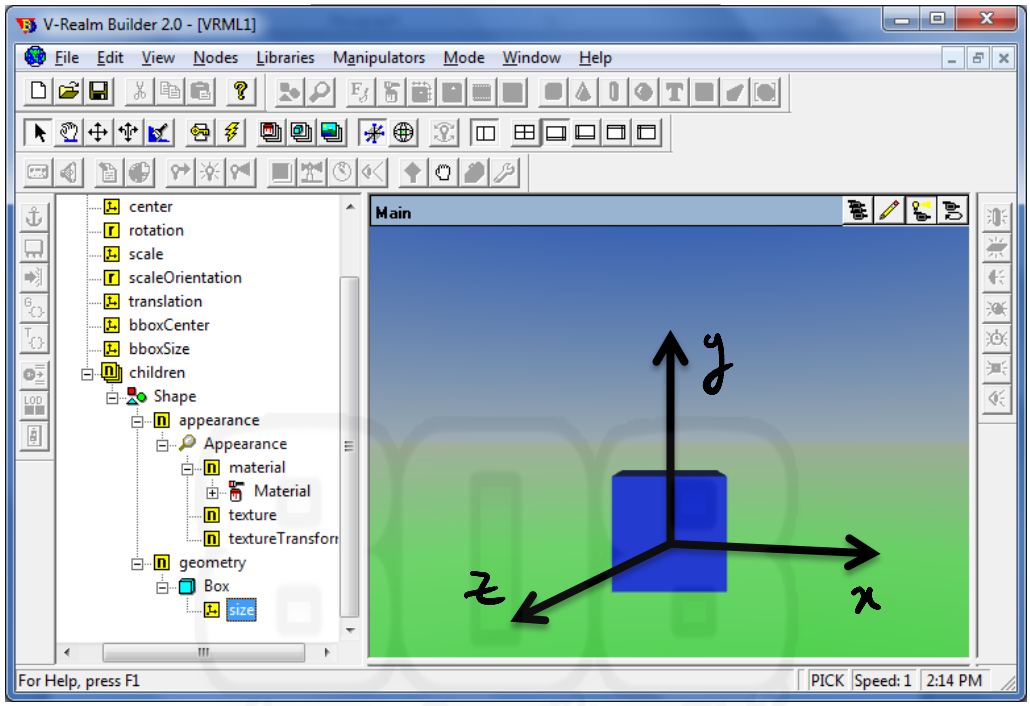
شکل ۱۰-۳ جعبه کنار x –axis را انتخاب نمایید

دیدگاه مولفین : با توجه به تصاویر سیاه و سفید، مولفین سعی کردند یک نمای کلی از محیط نرم افزار و مکان آیکن ها را نشان دهند.صحنه مجازی و محیط رنگی نرم افزار براحتی می تواند با دنبال کردن مرحله به مرحله گام های مطرح شده در بخش ۳-۳ برای خوانندگان گرامی میسر شود.

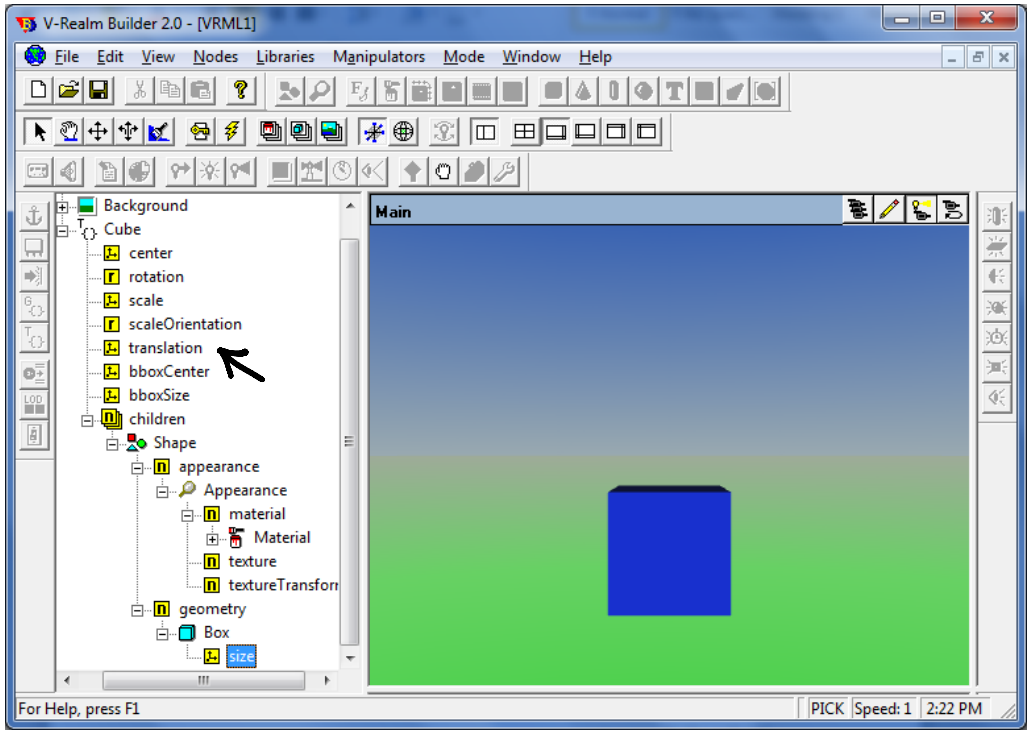
٤-٣. مختصات کارتزین برای VRML

صفحه صحنه مجازی دارای دو بعد می باشد،در VRML ، جهت مثبت محور x ، افقی و به سمت راست می باشد.جهت مثبت محور y ، عمودی و به سمت بالا می باشد.و جهت مثبت محور z ، عمود بر صفحه مانیتور (صفحه صحنه مجازی) و به سوی کاربر می باشد.(شکل ١١-٣) .

برای مشاهده قرارداد محورهای مختصات کارتزین نرم افزار VRML ، برروی خصوصیت translation شی گرافیکی Cube ، دابل کلیک کنید (شکل ١٢-٣) ، جعبه کنار x-Axis را تیک زده و با استفاده از نوار لغزنده ، مقدار آن را تغییر دهید. همانطور که مشاهده می کنید برای مقادیر مثبت x مکعب به سمت راست حرکت می کند. تیک x-Axis را زده تا از حالت انتخاب درآید و بروی تیک y-Axis کلیک کنید تا انتخاب شود و با استفاده از نوار لغزنده مقدار آن را تغییر دهید. همانطور که مشاهده می کنید برای مقادیر مثبت y ، مکعب به سمت بالا حرکت می کند. بار دیگر تیک y-Axis را زده تا از حالت انتخاب درآید و تیک z-Axis را بزنید تا انتخاب شود و با استفاده از نوار لغزنده ، مقدار z آن را تغییر دهید.همانطور که مشاهده می کنید برای مقادیر مثبت z مکعب به کاربر نزدیکتر می شود.



شکل ۱۱-۳ مختصات کارتزین برای VRML



شکل ۱۲-۳ خصوصیت translation

۳-۵. ایجاد mfile برای حل معادلات دیفرانسیل حاکم بر انتقال مکعب

برای حل معادله دیفرانسیل داده شده در معادله ۱-۳ ، از تابع ode45 برای حل انتگرال عددی معادله دیفرانسیل، استفاده می کنیم. برای کسب اطلاعات بیشتر در رابطه با تابع ode45 در پنجره دستور متلب عبارت help ode45 را تایپ کنید.

Mfile ای که در ادامه آورده شده ، cube_solver.m ، از معادله دیفرانسیل رابطه ۱-۳ ، به صورت عددی انتگرال می گیرد. عبارات توضیحی هم در این برنامه آورده شده تا توضیح مختصری در باره کارکرد هر قسمت برنامه داده شود.

```
%This is the m-script file for the example
Translating Cube:
% initial time:
t0=0;
% final time:
tf=10;
% Number of samples (for numerical
integration of the solution)
N=50;
% initial z position:
z_0=0;
% initial z velocity:
z_dot_0=0;
% Compute the solution of the differential
equation
[t,z]=ode45(@myfunc,[t0:(tf-t0)/N:tf],[z_0
z_dot_0]);
%plot the displacement

plot(t,z(:,1));
```

```
xlabel('Time[sec]')

ylabel('z[m]')
```

Mfile بالا ، cube_solver.m ، mfile تابعی myfunc.m را فراخوانی می کند تا بردار مشتق $[\dot{z}(t) \quad \ddot{z}(t)]$ را محاسبه کند. Mfile تابعی myfunc.m در ادامه آورده شده است.

```
function dzdt=myfunc(t,z)
%%mass
m=1;
%Force
F=-1;
%z_dot
dzdt_1=z(2);
%z_dot_dot
dzdt_2=F/m;
%derivative vector
dzdt=[dzdt_1;dzdt_2];
```

٦-٣. ایجاد mfile برای متحرک سازی صحنه مجازی

در ادامه ، صحنه مجازی ایجاد شده Cube_Virtual.wrl ، بر اساس راه حل داده شده توسط cube_solver.m متحرک سازی می شود. (خواننده باید cube_solver.m را قبل از cube_animate.m اجرا کند) . و انیمیشن ایجاد شده در یک فایل ویدیویی Cube_Movie.avi

ضبط خواهد شد. تنها پارامتری که در دنیای مجازی تغییر پیدا می کند خصوصیت جابجایی Translation مکعب می باشد. در نگاه اول شاید برنامه cube_animate.m مشکل بنظر برسد ولی با توضیحاتی که ارائه خواهد شد خواهید دید که چندان هم مشکل نیست !

برنامه نوشته شده قبل از حلقه for ، از دو بخش تشکیل شده است . قسمت اول آن Part 1 آن اشاره به باز کردن و دیدن دنیای مجازی دارد. و قسمت دوم آن Part 2 ، اشاره به تنظیمات ضبط انیمیشن ایجاد شده در یک فایل ویدیویی *.avi دارد. در این مثال فیلم با نام Cube_Movie.avi ذخیره می شود.

همان گونه که قبلا در این بخش گفته شد ، تنها پارامتری که در دنیای مجازی تغییر می کند خصوصیت جابجایی مکعب می باشد، صحنه مجازی ایجاد شده در بخش یک برنامه ، world نامگذاری می شود.

```
world = vrworld('Cube_Virtual');
```

برای تغییر خصوصیت جابجایی شی گرافیکی مکعب ، کد زیر را بنویسید :

```
world.Cube.translation = [0 0 z(Index,1)];
```

بعد از اینکه شما cube_animate.m را اجرا کردید ، فایل ویدیویی Cube_Movie.avi در مسیر جاری نرم افزار متلب ایجاد خواهد شد.

```
% creates a virtual world associated with
Cube_Virtual.wrl file
world = vrworld('Cube_Virtual');
%open the virtual world
open(world);
%draw the virtual world
fig = view(world, '-internal');
%defining the interval time to record the
animation
set(world, 'RecordInterval', [0 tf]);
%defining the name of the movie file to be
cube.avi
set(fig,
'Record2DFileName','Cube_Movie.avi');
% enabling 2D recording
set(fig, 'Record2D', 'on');
%compression quality of the movie
set(fig, 'Record2DCompressQuality', 100);
%defining recording mode to be scheduled
set(world, 'RecordMode', 'scheduled');
%initiating the counter
Index=0;
%sweeping the time interval
for t=t0:(tf-t0)/N:tf
    %incrementing the counter
    Index=Index+1;
    %changing the translation property of the
object Cube that was defined
    %in Cube_Virtual.wrl to the value
computed by ode45
    world.Cube.translation=[0 0 z(Index,1)];
```

```
    %changing the time of the virtual world
to match the time that was

    %given by ode45
    set(world,'Time', t);

    %update the virtual reality scene
    vrdrawnow;
end
%close the virtual world
close(world);
delete(world);
```

مراجع

کتاب

Hibbler RC (1986) Engineering mechanics: dynamics, 4th edn. McMillan, New York