

Nassim Khaled

# Virtual Reality and Animation for MATLAB<sup>®</sup> and Simulink<sup>®</sup> Users

Visualization of Dynamic Models and  
Control Simulations

 Springer

# Virtual Reality and Animation for MATLAB<sup>®</sup> and Simulink<sup>®</sup> Users



Nassim Khaled

# Virtual Reality and Animation for MATLAB<sup>®</sup> and Simulink<sup>®</sup> Users

Visualization of Dynamic Models  
and Control Simulations



Springer

Nassim Khaled  
Cummins Technical Center  
1900 Mckinley Avenue  
Columbus, IN 47201  
USA  
nassim.khaled@gmail.com

ISBN 978-1-4471-2329-3 e-ISBN 978-1-4471-2330-9  
DOI 10.1007/978-1-4471-2330-9  
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data  
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2011943558

© Springer-Verlag London Limited 2012

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

# Contents

<b>1 Introduction to Virtual Reality in MATLAB® and Simulink® Environment .....</b>	<b>1</b>
1.1 Overview of the Book .....	1
<b>2 V-Realm Builder® .....</b>	<b>3</b>
2.1 Introduction.....	3
2.2 What Is V-Realm Builder?.....	3
2.3 Installing V-Realm Builder from MATLAB® .....	3
2.4 Starting V-Realm Builder.....	4
<b>3 The Translating Cube .....</b>	<b>7</b>
3.1 Introduction.....	7
3.2 Cube Problem.....	8
3.3 Creating the Virtual Scene for the Cube .....	8
3.4 Cartesian Coordinates for the VRML .....	14
3.5 M-script File for Solving the Differential Equation.....	15
3.6 M-script File for Animating the Virtual Scene .....	16
3.7 Application Problem .....	18
Reference .....	19
Book .....	19
<b>4 Mass-Spring-Damper Oscillations .....</b>	<b>21</b>
4.1 Introduction.....	21
4.2 Mass-Spring-Damper Problem .....	22
4.3 Creating the Virtual Scene for the Mass-Spring-Damper.....	22
4.4 Adding Material and Changing the Color Properties of the Virtual Objects .....	26
4.5 M-script File for Solving the Differential Equation.....	28
4.6 M-script File for Animating the Virtual Scene .....	31
4.7 Application Problem .....	33
Reference .....	34
Book .....	34

<b>5 Crank-Slider Mechanism of a Piston .....</b>	<b>35</b>
5.1 Introduction.....	35
5.2 Crank-Slider Mechanism .....	35
5.3 Creating the Virtual Scene for the Crank-Slider Mechanism .....	36
5.4 M-script File for Animating the Virtual Scene .....	42
5.5 Application Problem: Control of a Ball on a Plate .....	46
References.....	48
Books.....	48
<b>6 Car Animation with Joystick Control.....</b>	<b>49</b>
6.1 Introduction.....	49
6.2 Equations of Motion of the Car and Wheels.....	50
6.3 Creating the Virtual Scene for the Car.....	52
6.4 Joystick Setup .....	67
6.5 M-script File for Animating the Virtual Scene .....	67
6.6 Changing the Viewpoint.....	71
6.7 Application Problem: Fuzzy Logic Controller for the Car Speed.....	72
Reference .....	74
Book.....	74
<b>7 Animation of a Ship Moving Across Waves.....</b>	<b>75</b>
7.1 Introduction.....	75
7.2 Equations of Motion of Sea Waves and the Ship.....	75
7.3 Creating the Virtual Scene for the Sea and Ship.....	76
7.4 M-script File for Animating the Virtual Scene .....	93
7.5 Application Problem: Ship Heading Controller.....	95
References.....	98
Books.....	98
<b>8 Animation of a Translating Cube Using Simulink® .....</b>	<b>99</b>
8.1 Introduction.....	99
8.2 Translating Cube Problem .....	99
8.3 Cartesian Coordinates for the Virtual Reality Environment VRML .....	100
8.4 Creating the Simulink® Model for Animating the Virtual Scene .....	101
8.5 Application Problem: Collision of Two Boxes .....	106
Reference .....	108
Book.....	108
<b>9 Animation of Mass-Spring-Damper Oscillations     Using Simulink® .....</b>	<b>109</b>
9.1 Introduction.....	109
9.2 Mass-Spring-Damper Problem .....	109
9.3 Creating the Simulink® Model for Animating the Virtual Scene .....	110

- 9.4 Changing the Viewpoint in Simulink® ..... 118
- 9.5 Application Problem: Collision of Two Boxes ..... 119
- Reference ..... 121
  - Book ..... 121
- 10 Animation of Crank-Slider Mechanism of a Piston**
- Using Simulink®** ..... 123
- 10.1 Introduction ..... 123
- 10.2 Crank-Slider Mechanism ..... 123
- 10.3 Creating the Simulink® Model for Animating  
the Virtual Scene ..... 124
- 10.4 Changing the Viewpoint in Simulink® ..... 128
- 10.5 Application Problem: Control of a Ball on a Plate ..... 129
- References ..... 139
  - Books ..... 139
- 11 Car Animation with Joystick Control for Simulink® Users** ..... 141
- 11.1 Introduction ..... 141
- 11.2 Equations of Motion of the Car and Wheels ..... 142
- 11.3 Joystick Setup ..... 144
- 11.4 Creating the Simulink® Model for Animating  
the Virtual Scene ..... 146
- 11.5 Application Problem: Fuzzy Logic Controller  
for the Car Speed ..... 149
- Reference ..... 160
  - Book ..... 160
- 12 Animation of a Ship Moving Across Waves Using Simulink®** ..... 161
- 12.1 Introduction ..... 161
- 12.2 Equations of Motion of Sea Waves and the Ship ..... 161
- 12.3 Simulink® Model for Animating the Virtual Scene ..... 162
- 12.4 Application Problem: Ship Heading Controller ..... 169
- References ..... 172
  - Books ..... 172
- Index** ..... 173





# Chapter 1

## Introduction to Virtual Reality in MATLAB® and Simulink® Environment

### 1.1 Overview of the Book

Whether the reader wants to model the dynamics of a simple pendulum or design a controller for a robot arm, it is always useful to be able to animate the dynamic model. Animation of a dynamic model enables one to observe all the degrees of freedom of a system at once, rather than looking at 2D and 3D plots of simulated data. Such animations serve as a debugging tool for developing the dynamic model in hand. It can also help in preliminary assessment of the performance of the controller. In addition, animations can help students and less-technical audience to understand the end results of a dynamic model without going into lengthy mathematical derivations.

Two of the top leading tools in dynamic modeling and control development are MATLAB® and Simulink®. Researchers and the industry have been increasingly using MATLAB® and Simulink® in the past two decades. One would be surprised to know that many of M-script file users are not comfortable using Simulink® and vice versa. Thereby the author has decided in this book to address both users separately. For both sets of users, the first step to create a virtual reality animation is to draw the object(s) (along with the virtual scene) in V-Realm Builder, and the second is to use MATLAB® M-script or Simulink® model to vary the states or properties of the object(s) and/or the scene based on a set of differential equations, prescribed trajectories, or empirical formulas.

The main theme of this book along with the supplementary electronic material is to “learn through application.” Step-by-step examples in each chapter with illustrative screen shots will guide the reader through this learning experience. Furthermore, users can download the virtual scenes, the movie recordings for the animations, and the M-script files (or the Simulink® models) for the examples included in each chapter from Springer’s web site by following the link <http://extras.springer.com/>

---

Matlab® and Simulink® are registered trademark of The Mathworks, Inc.

and searching for the ISBN of the book. These downloadable materials should help users understand the end result of each example, explore the models, and debug their own models.

The author tried to keep the chapters as independent as possible. If the reader is not interested in becoming an expert in virtual reality, this introduction will help him/her skip to his/her desired application. It is still strongly advised to go through all the chapters in order to gain a good level of experience in virtual reality.

Chapter 1 of this book is a short introduction to the book and its contents. Chapter 2 describes the main tool, V-Realm Builder, which is used to set up and draw the virtual reality objects and the virtual environment. Chapters 3, 4, 5, 6, and 7 are intended for M-script file users. They contain step-by-step examples with detailed screen shots on how to create an object in V-Realm Builder and how to write an M-script file to link the dynamic model (that is governed by equation(s) of motion, empirical formula, or prescribed trajectory) to the virtual scene in order to animate the object(s) and/or the surrounding scene. Chapters 3, 4, 5, 6, and 7 are sorted in ascending level of difficulty. Each chapter includes the utilization of new properties and features of V-Realm Builder. Chapters 3, 4, 5, 6, and 7 contain the following examples, respectively:

- The translating cube in addition to an application problem of the collision of two objects
- The mass-spring-damper oscillations in addition to an application problem of two masses connected by a spring
- The crank-slider mechanism of a piston in addition to an application problem of PID control of a ball on a plate
- Car animation with joystick control in addition to an application problem of fuzzy logic control of the speed of the car
- Animation of a ship moving across waves in addition to an application problem of fuzzy logic control of the heading of the ship

Chapters 8, 9, 10, 11, and 12 contain a step-by-step guide with detailed screen shots to animate the virtual worlds constructed in Chaps. 3, 4, 5, 6, and 7 using Simulink® in addition to the application problems presented. Finally, a short index to the supplementary material of the chapters is covered at the end of the book. The supplementary material includes movie recordings for all the problems covered in the book and M-script files and Simulink® models that were used to animate the virtual systems in addition to all the virtual scenes that were constructed in Chaps. 3, 4, 5, 6, and 7. The supplementary material can be downloaded from Springer's web site, <http://extras.springer.com/>, by searching for the ISBN of the book.

Few comments regarding Matlab® version and the operating system:

- the codes of this book have been developed using Matlab® 2008a and windows 7 operating system and checked for compatibility using Matlab® 2011b.
- as of Matlab® 2009a, **Virtual Reality Toolbox** has been renamed **Simulink 3D Animation** thus users who will be using Matlab® 2009a or later should know that the only difference, in the context of this book, is the name of the toolbox.

# Chapter 2

## V-Realm Builder<sup>®</sup>

### 2.1 Introduction

Section 2.2 of this chapter gives the user a brief outline of V-Realm Builder Software<sup>®</sup> which will be used in Chaps. 3, 4, 5, 6, and 7 to construct the virtual objects and scenes.

Section 2.3 guides the user through the installation process of the V-Realm Builder Software<sup>®</sup>.

Section 2.4 helps the user start V-Realm Builder Software<sup>®</sup>.

### 2.2 What Is V-Realm Builder?

V-Realm Builder (which in the context of this book will sometimes be referred to as VRML) is a software that is used to design virtual worlds and draw/import 3D virtual objects. After the user builds his/her virtual world, he/she can manipulate the objects and the virtual scene through MATLAB<sup>®</sup> commands and Simulink<sup>®</sup> models to animate the scene. Virtual 3D objects' properties such as translation, rotation, scale, and color can be changed through MATLAB<sup>®</sup> commands and Simulink<sup>®</sup> models.

### 2.3 Installing V-Realm Builder from MATLAB<sup>®</sup>

The user needs to purchase the MATLAB<sup>®</sup>'s Virtual Reality Toolbox<sup>®</sup> to have access to V-Realm Builder. To check if Virtual Reality Toolbox<sup>®</sup> is installed, the following command should be typed in the MATLAB<sup>®</sup> command window:

***vrinstall -check***

---

V-Realm Builder 2.0 is copyrighted, 1996-1997, by Integrated Data Systems, Inc. Matlab<sup>®</sup> is a registered trademark of The Mathworks, Inc.

If the V-Realm Builder is installed correctly, the following statement will be returned in the MATLAB® command window:

***VRML viewer: installed***

***VRML editor: installed***

If VRML viewer or editor is not installed, you can reinstall them both by typing the following in the MATLAB® command window:

***vrinstall -install***

MATLAB® will display the following message in the command window:

***Installing blaxxun Contact viewer...***

***Do you want to use OpenGL or Direct3D acceleration? (o/d)***

The user should type **o** (or **d**) and hit **Enter** or **Return** on the keyboard and follow the screen instructions to finish the installation.

If MATLAB® command window returns the following:

***??? Undefined command/function 'vrinstall.'***,

Then the user does not have the Virtual Reality Toolbox®. It can be purchased by contacting a sales representative for MathWorks by the following link: <http://mathworks.com/>.

## 2.4 Starting V-Realm Builder

In order to start V-Realm Builder, the user has to go to the location where MATLAB® was installed and go to the following subdirectory location:

***{Matlab installation folder}\toolbox\vr\vrealm\program***

In other words, if MATLAB® was installed in **C:\Matlab**, then the user can access V-Realm Builder by going to the following directory (inside the rectangle in Fig. 2.1):

***C:\Matlab\toolbox\vr\vrealm\program***

When the user goes to the right subdirectory, he/she will see V-Realm Builder executable, which is named ***vrbuild2*** (inside the ellipse in Fig. 2.1).

If the user is going to use V-Realm Builder frequently, it is easier to create a shortcut on the desktop or any other convenient location for ***vrbuild2*** rather than accessing the directory every time. To launch V-Realm Builder, the user should double-click ***vrbuild2***, and the main window of V-Realm Builder will open (Fig. 2.2).

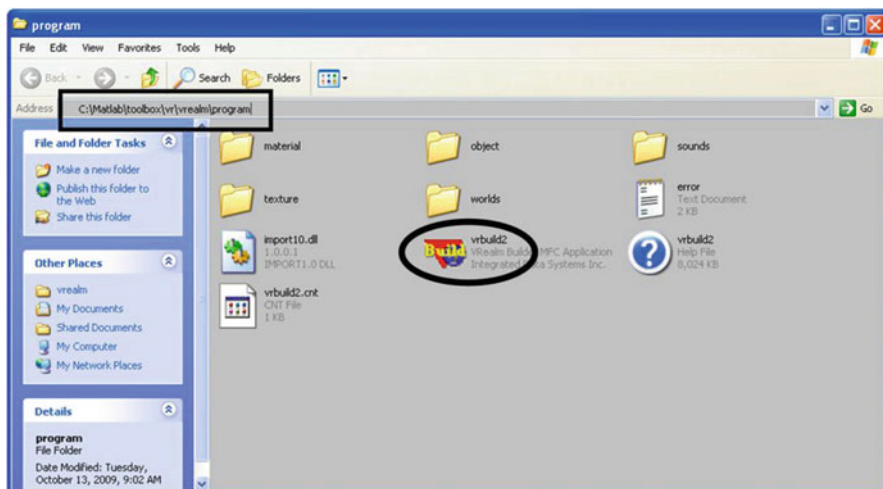


Fig. 2.1 Directory location of V-Realm Builder

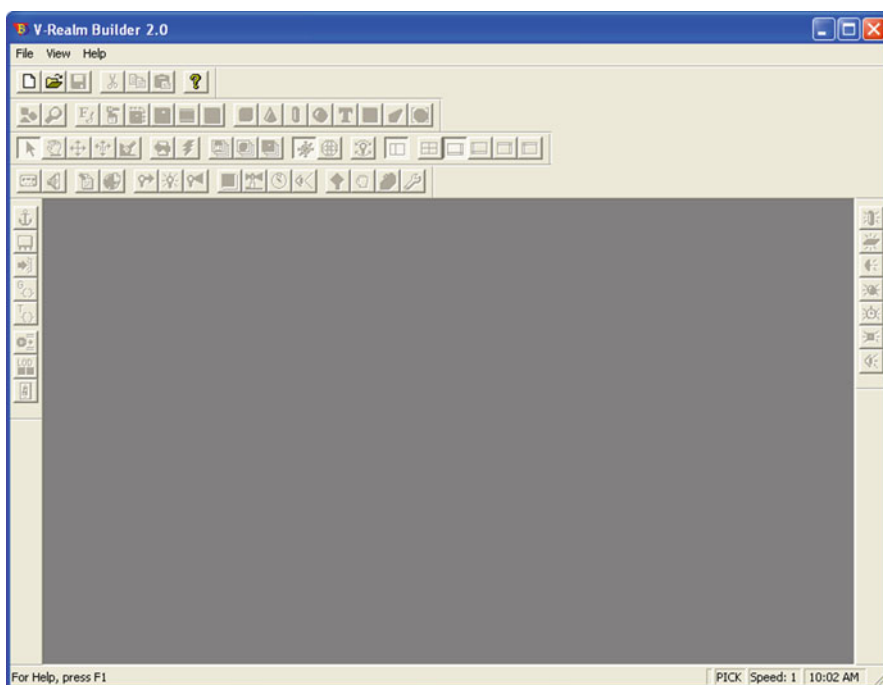


Fig. 2.2 Main window of V-Realm Builder

# Chapter 3

## The Translating Cube

### 3.1 Introduction

The purpose of this chapter is to introduce MATLAB® users that are interested in animating their physical system in a virtual reality environment to the virtual reality tool, V-Realm Builder (VRML). This chapter will walk the new user through a complete exercise on how to animate a physical problem governed by an equation(s) of motion in VRML environment using M-script commands.

In Sect. 3.2, a simple dynamics problem of a cube that is subjected to an external force will be presented. Newton's second law will be used to derive the equation of motion.

In Sect. 3.3, step-by-step instructions will guide the reader through drawing the cube in VRML and changing its name, color, and size. A simple background will also be added to the scene.

The  $x$ -,  $y$ -, and  $z$ -axes convention of VRML will be briefly explained in Sect. 3.4. In addition, a simple familiarization experiment will be described to enable the reader to better understand the coordinate system of VRML.

In Sect. 3.5, an M-script file that numerically computes the solution of the differential equation (derived from Newton's second law) will be developed.

To animate the virtual scene, an M-script file that uses the numerical solution from Sect. 3.5 will be described in Sect. 3.6. Furthermore, a procedure to save the animation as a movie file will be included.

The chapter concludes by an application problem of two colliding masses in Sect. 3.7.

The electronic version of all the M-script files and VRML models in addition to the recorded movies for the cube and the collision problem can be downloaded from Springer's web site <http://extras.springer.com/>.

---

Matlab® is a registered trademark of The Mathworks, Inc.

## 3.2 Cube Problem

Given a cube of mass  $m = 1(\text{kg})$  and of dimensions  $0.5 \times 0.5 \times 0.5(\text{m}^3)$ . The cube is subjected to the following external force:  $F_z = -1(\text{N})$ .

Using Newton's second law, the following second-order differential equation for the  $z$ -displacement (check Sect. 3.4 for coordinate's convention) of the cube can be developed:

$$m\ddot{z}(t) = F_z \Rightarrow \ddot{z}(t) = \frac{F_z}{m} \quad (3.1)$$

The purpose of this exercise is to animate the solution of this equation for zero initial conditions.

## 3.3 Creating the Virtual Scene for the Cube

To create the virtual scene for the cube, follow the following steps:

1. Open the V-Realm Builder and click on **File>New** to create a new virtual world.
2. Click on the **Insert Background** button (Fig. 3.1) to create a background for the scene.
3. Click on the **Insert Box** (Fig. 3.2) to create the cube. In the main view window, you should see a cube with a green and blue background behind it. Notice that on the left side of the main view window, some text and yellow icons have appeared, all under the title **Transform** (Fig. 3.3).
4. Click once on the title **Transform** to select it. Click another time on **Transform** and change the text to a meaningful name such as **Cube** (Fig. 3.4). Do not double-click on **Transform**; you should wait at least 1 s after clicking the first time. The purpose behind changing the name from **Transform** to **Cube** is to allow MATLAB<sup>®</sup> to uniquely identify the object **Cube**. In other words, suppose you wanted to have two cubes instead of one, then both of the cubes will have the name **Transform**, and MATLAB<sup>®</sup> will not be able to differentiate between both objects. In this case, you better name one cube as **Cube1** and the other as **Cube2**.
5. Click on the **Color Mode** button (Fig. 3.5).
6. Double-click on the **Diffusive Color** gray area (Fig. 3.6).
7. Click on the blue color or any other color you prefer (Fig. 3.7). Press OK when done.
8. Click on any face of the cube, and the whole object will turn blue (Fig. 3.8). Press Close when done.
9. By default, the size of the cube is  $2 \times 2 \times 2$  and its center is positioned at the origin  $(0,0,0)$ . To change its size, click the + sign to the left of **Box** and then double-click on **size** (Fig. 3.9). Tick the  $x$ -axis box (Fig. 3.10) and change



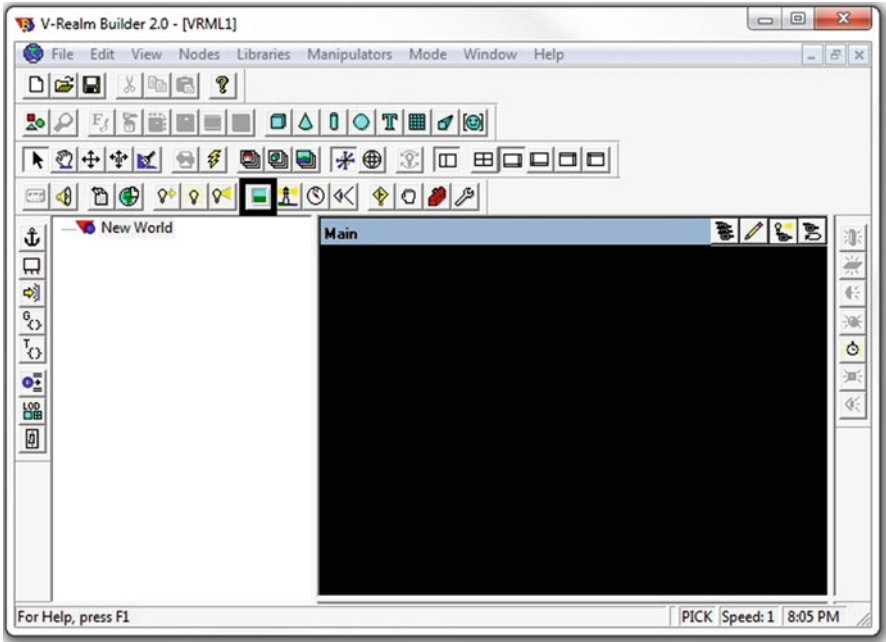


Fig. 3.1 Insert Background button is shown inside the rectangle

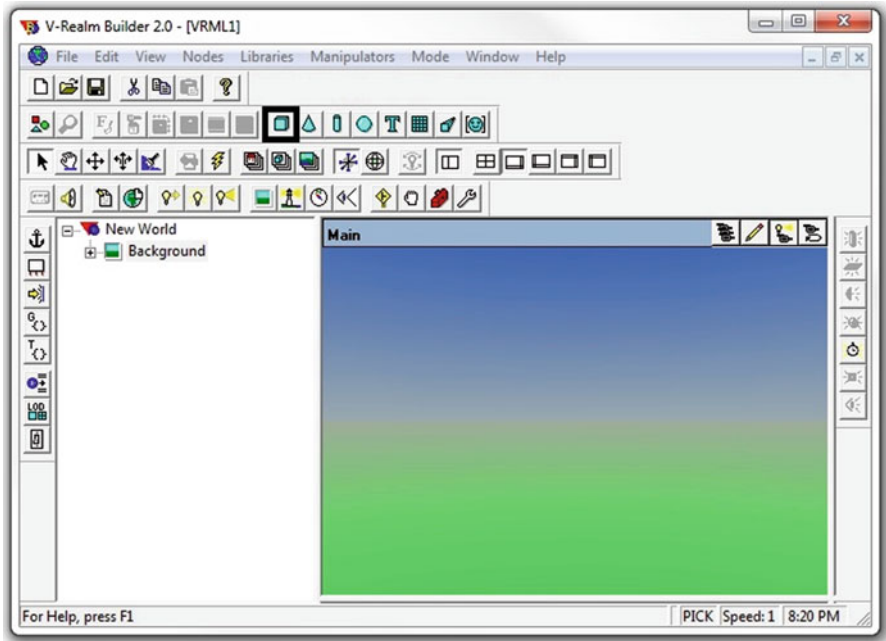


Fig. 3.2 Insert Box button is shown inside the inside the rectangle

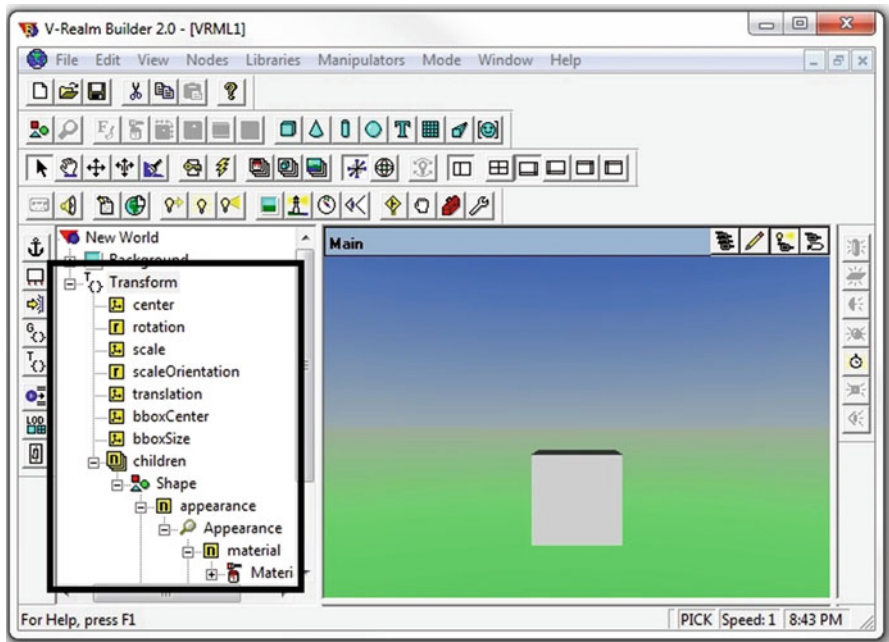


Fig. 3.3 The text and the yellow icons appear under the title Transform

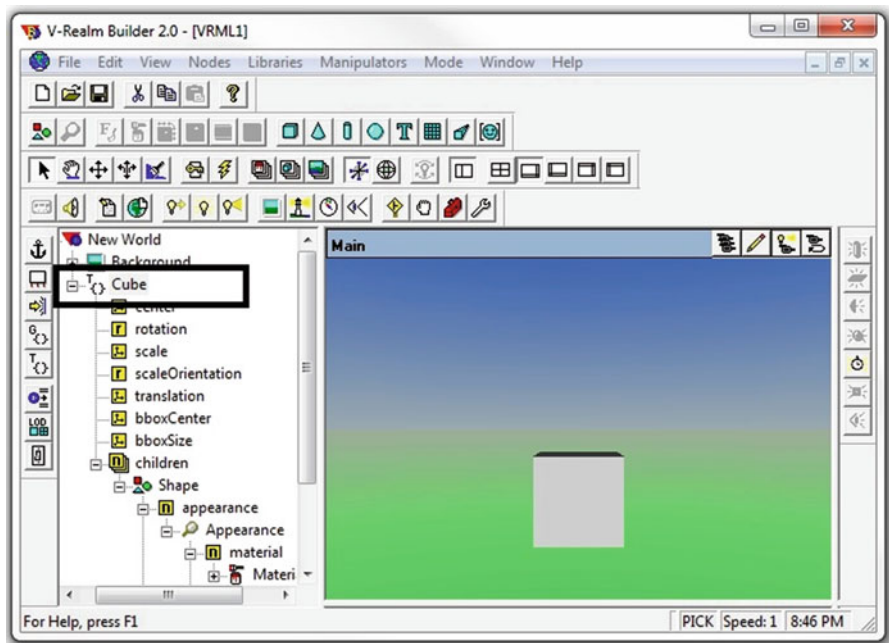


Fig. 3.4 Change the name of the cube from Transform to Cube