

SONY®

Community Place™ Conductor 2.0 **ユーザーズマニュアル**



警告

ソフトウェア製品は、安全のための注意事項を守らないと、健康を害することがあります。

この取扱説明書には、健康を守るための重要な注意事項と製品の取り扱いかたを示してあります。この取扱説明書をよくお読みのうえ、製品をお使いください。お読みになったあとは、いつでも見られるところに必ず保管してください。

下記の注意事項をよくお読みください。製品全般の注意事項が記されています。

警告表示の意味

インストールガイドおよび製品では、次のような表示をしています。表示の内容をよく理解してから本文をお読みください。



この表示の注意事項を守らないと、健康を害するおそれがあります。

この行為を禁止する記号



禁止



下記の注意を守らないと
健康を害するおそれがあります。



禁止

・ディスプレイ画面を長時間継続して見ない

ディスプレイなどの画面を長時間継続して見続けると、目が疲れたり、視力が低下するおそれがあります。ディスプレイ画面を見続けて体の一部に不快感や痛みを感じたときは、すぐにコンピューターの使用をやめて休息してください。万一、休息しても不快感や痛みがとれないときは、医師の診察を受けてください。



禁止

・キーボードを使いすぎない

キーボードやマウスなどを長時間継続して使用すると、腕や手首が痛くなったりすることがあります。キーボードやマウスなどを使用中、体の一部に不快感や痛みを感じたときは、すぐにコンピューターの使用をやめて休息してください。万一、休息しても不快感や痛みがとれないときは、医師の診察を受けてください。

Copyright © 1998 Sony Corporation. All rights reserved.
Community Place™ はソニー株式会社の商標です。

このソフトウェアの使用および複製は、契約書の条項で許容されている範囲でのみ許可されています。本書の記述内容は、予告なく変更する可能性があり、また、製品仕様またはソニー株式会社の義務遂行の確約を表すものではありません。ソニー株式会社の書面による許可なく、この出版物の一部または全部を再生産、転送、転写、保管、または他言語へ翻訳することは、その形態に関係なく、一切禁じられています。

Windows®、Windows NT® は米国 Microsoft Corporation の米国及びその他の国における登録商標です。

Java™、JDK™ は、米国およびその他の国における米国サン・マイクロシステムズ社の商標です。

Netscape Navigator™ は米国 Netscape Communications Corporation の商標です。
その他、記載されている会社名、製品名は、各社の商標または登録商標です。

目次

Community Place Conductor について	1
インストール	2
アンインストール	3
Conductor の起動と終了	3
Conductor の各ウィンドウの機能	4
リファレンス	5
 第1章 Main ウィンドウ	6
File メニュー	6
New	6
Open、Reopen	6
Save	7
Save As	7
Close	7
Exit	7
Edit メニュー	7
Undo	7
Redo	7
Cut	8
Copy	8
Paste	8
Delete	8
Keyframe	8
View メニュー	8
Run メニュー	8
Run	8
Pause	9
Stop	9
Script メニュー	9
Expert	9
Build	9
Rebuild	9
Options メニュー	9

Environment	9
Help メニュー	12
VRML97 Spec	12
Show CLASSPATH	13
About	13
マウスモード	14
ツールバー	14
第2章 3D Perspective View ウィンドウ	17
VRML ファイルの読み込み	17
オブジェクトの選択	18
視点位置と視線方向の表示	18
オブジェクトの生成	18
ナビゲーションの方法	19
オブジェクトの移動	20
オブジェクトの回転	20
オブジェクトのスケーリング	20
ポップアップメニュー	21
第3章 Appearance ウィンドウ	24
Material ノードの編集	24
Texture ノードの編集	25
TextureTransform ノードの編集	26
第4章 Attribute ウィンドウ	27
DEF 名の編集	28
属性の編集	28
Wizard タブ	29
PROTO 内のノード編集	30
Script ノードに対する編集	31
第5章 Keyframe Editor ウィンドウ	32
Keyframe Editor の起動および終了方法	32
Source の選択	33

表示の拡大縮小、スクロール	33
Fraction の設定	33
表示のロック	33
アニメーションの動作確認	33
Loop チェックボックス	34
キーフレームの複数選択	34
ポップアップメニュー	34
キーフレームの操作 (スケーリング、スムージング)	35
第 6 章 Library ウィンドウ	38
ドラッグ&ドロップ操作	39
関連アプリケーションの起動	39
Script ライブラリのヘルプ表示	40
ライブラリフォルダーの変更	40
アイコンのカスタマイズ	41
各ライブラリでの対応フォーマット	41
ライブラリへの追加	42
第 7 章 Message ウィンドウ	43
Warning メッセージ	43
Java コンパイラからのメッセージ	43
第 8 章 Multi User ウィンドウ	44
マルチユーザーワールドの作成	44
マルチユーザーサーバーの設定	44
使用アバターの指定	45
マルチユーザーワールドの動作確認	49
第 9 章 Parallel View ウィンドウ	50
視点位置と視線方向の表示	50
ポップアップメニュー	51
オブジェクトの生成	52
ナビゲーションの方法	52

オブジェクトの移動	53
オブジェクトの回転	53
オブジェクトのスケーリング	53
第 10 章 Route ウィンドウ	54
リスト表示での操作	54
ビジュアル表示での操作	56
DEF 名の自動生成と変更	58
ポップアップメニュー	58
第 11 章 Scene Graph ウィンドウ	60
ファイルスコープの切り替え	60
PROTO スコープの切り替え	61
ノードの選択	62
ヘルプの表示	62
ポップアップメニュー	62
第 12 章 Script Expert ウィンドウ	67
Script の生成法	67
第 13 章 Text Editor ウィンドウ	69
Text Editor によるファイルの開き方	69
Text Editor の終了方法	69
ポップアップメニュー	70
第 14 章 Texture Mapping ウィンドウ	72
Color ページ	72
Texture Coordinate ページ	75
Change Color Mode ダイアログボックス	78
Change Texture Mode ダイアログボックス	81
Texture Mapping Projection ダイアログボックス	83
第 15 章 Object Customize Wizard の作成方法	85

VRML オブジェクトの変更	86
Wizard の設定ファイルの記述	86
スクリプトリファレンス	92

Community Place Conductor について

Community Place™ Conductor 2.0 (以下、Conductor と呼びます。)は、VRML97 に対応したオーサリングツールです。

Conductor には、以下のような特長があります。

VRML97 対応

ドラッグ&ドロップによるノードの追加

ノード属性の数値入力機能

音、物体のテクスチャーの定義が可能

実行モードと編集モードの切り替えが可能

Java™ 対応 (JRE1.1.3 以降あるいは、JDK1.1.3 以降が必要です。)

スコープの切り替え、スコープ単位の編集が可能

スクリプトノードを自動作成できるスクリプトエキスパート機能

ビジュアルなルーティング表示、追加、削除

キーフレームアニメーション (Interpolator) 編集機能

PROTO 編集機能

USE のサポート

このマニュアルでは個々の機能について解説します。Conductor を使用してワールドを作成する方法については、チュートリアルをご覧ください。

本文中のイタリック文字 (例 *color*) は、VRML のフィールド名を、ボールド文字 (例 **value=32**) は、プログラムの一部を示しています。詳しくは、巻末に挙げる参考文献をご覧ください。

■ インストール

Conductor を実行するには、システムが以下の条件を満たしているかどうか確認してください。

表 1 使用環境

	推奨環境
Hardware	Windows® 95/98、Windows NT® 4.0 以上の稼動している PC/AT 互換機
CPU	Pentium® 90MHz 以上
メモリー	32MB 以上
空きディスク容量	20MB 以上（インストール時には 40MB 必要）
画面解像度	800x600 ピクセル以上
表示色数	65536 色以上
ソフトウェア	Netscape Navigator™ for Windows (ver 3.0 以降) または、Microsoft® Internet Explorer (ver 3.0 以降) (注 1)
JAVA™ (注 2)	JDK1.1.3 以降
サウンド	サウンドボード + スピーカーまたはヘッドホン

(注 1) ヘルプの表示や、Anchor の表示を行う際に必要です。

(注 2) Java を使用したコンテンツの作成を行う場合に必要です。

(注) 一部のハードウェアでは正しく動作しない場合があります。詳しくは、オンラインの制限事項（Conductor をインストールしたフォルダの doc¥restrict¥restrict-j.txt）をお読みください。

インストールを行うには、setup.exe（CD-ROM を読み込んだ際に自動的に起動します）を起動してインストールを行ってください。

■ アンインストール

アンインストールするには、スタートメニューから、プログラム Community Place Conductor Uninstall を選択します。

(注) アンインストールしたあとも、インストール後作成したファイルや一時ファイルなどが残ることがあります。アンインストール実行後に Conductor のディレクトリが残っていたらすべて削除してください。Conductor を再インストールする際には、Conductor のディレクトリがすべて削除されていることを確認してから、インストールすることをお勧めします。

■ Conductor の起動と終了

• 起動方法

スタートメニューから、プログラム Community Place Conductor Community Place Conductor を選択します。

ワールドを新規に作成するには、File Newメニューを選択してください。既存の VRML97 ファイルを読み込むには、File Open メニューを選択してください。

(注) VRML1.0 には対応しておりません。ファイルが VRML1.0 の場合は、あらかじめ、Conducotr 2.0 CD-ROM の中に収められている vrml1to2 で、VRML2.0 に変換しておいてください。

• 終了方法

メニューから File Exit を選択するか、Main ウィンドウ右上の×ボタンをクリックすることで終了させることができます。ファイルが変更されている場合には、保存を促すダイアログが表示されます。

■ Conductor の各ウィンドウの機能

<u>Main ウィンドウ</u>	: Conductor 全体の管理を行います。
<u>3D Perspective View ウィンドウ</u>	: 透視投影で物体を表示します。
<u>Appearance ウィンドウ</u>	: オブジェクトの外観を設定します。
<u>Attribute ウィンドウ</u>	: ノードの属性を設定します。
<u>Keyframe Editor ウィンドウ</u>	: Interpolator の編集を行います。
<u>Library ウィンドウ</u>	: ライブラリの管理をします。
<u>Message ウィンドウ</u>	: 各種メッセージを表示します。
<u>Multi User ウィンドウ</u>	: マルチユーザーワールドを設定します。
<u>Parallel View ウィンドウ</u>	: 平行投影で物体を表示します。
<u>Route ウィンドウ</u>	: イベントのルーティングを設定します。
<u>Scene Graph ウィンドウ</u>	: ワールドのツリー構造を表示します。
<u>Script Expert ウィンドウ</u>	: スクリプトのひな形を作成します。
<u>Text Editor ウィンドウ</u>	: Java ファイルを編集する時に使います。
<u>Texture Mapping ウィンドウ</u>	: Texture Mapping の編集を行います。

各ウィンドウは必要に応じて表示したり、閉じておくことができます。

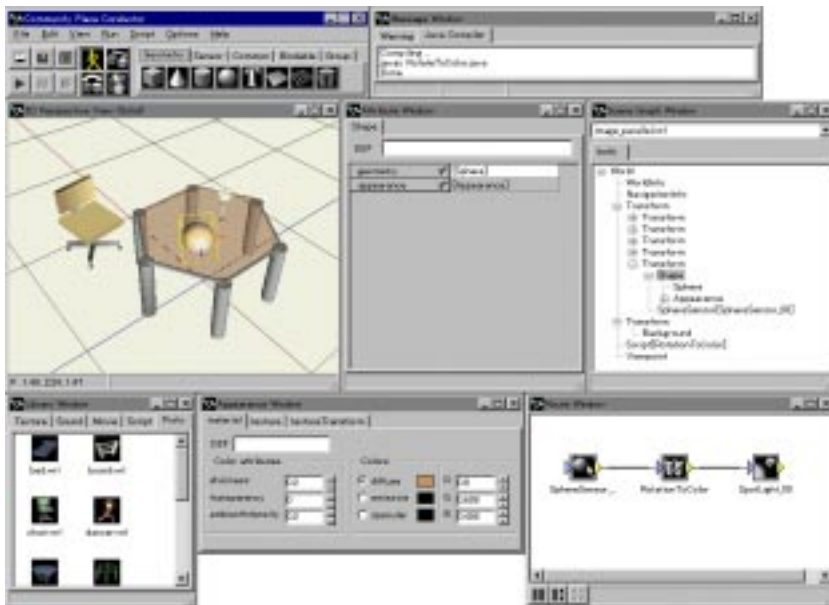


図 1 Conductor を使用している様子

■ リファレンス

VRML97 仕様

<http://vag.vrml.org/VRML97/>

Virtual Society on the Web ホームページ

<http://vs.sony.co.jp/>

第1章 Main ウィンドウ

図2に示す Main ウィンドウでは、ワールドの新規作成、読み込み、保存、実行、停止、マウスモードの切り替え、キーフレーム編集モードの切り替え、ノードの追加、削除などを行います。

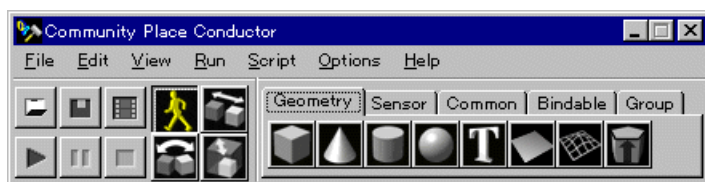


図2 Main ウィンドウ


■ File メニュー

● New

新規にワールドを作成します。

3D Perspective View ウィンドウ (以下 3D View ウィンドウ) などのワールドを操作するためのウィンドウが表示されます。

● Open、Reopen

VRML ファイルをロードします。  (Open ボタン) を押しても同じ動作をします。ファイルオープンダイアログが表示されますので、読み込みたいワールドのファイルを選択してください。


以前に読み込んだことのあるファイルを読み込むには、File メニューの Reopen でファイルの一覧が表示されますので、読み込むファイルを選んでください。

Open、Reopen の他にも、以下の方法でファイルを開くことができます。ただし、いずれの方法も VRML ファイルのショートカットには対応してません。

- Conductor のアイコンに VRML ファイルをドロップする。
- Conductor を起動しておいて、Main ウィンドウに VRML ファイルをドロップする。
- コマンドラインによりロードする。(例えば、cpcondct foo.wrl)

(注) VRML1.0 には対応していません。ファイルが VRML1.0 の場合は、あらかじめ、Conducotr 2.0 CD-ROM の中に収められている vrml1to2 で、VRML2.0 に変換しておいてください。

- **Save**

ワールドをファイルに保存します。  (Save ボタン) を押しても同じ動作をします。

- **Save As**

ワールドを別ファイルに保存します。ファイル保存のダイアログが表示されますので、ファイル名を入力してください。

- **Close**

編集集中のワールドを閉じます。

- **Exit**

Conductor を終了します。

ワールドが変更されている場合は、保存するか確認してきますので、保存する場合は「はい」を選択してください。

(注) 起動時、終了時にテンポラリディレクトリ (デフォルトは、
"C:¥ProgramFiles¥Sony ¥Community Place Conductor ¥tmp") 内にあるファイルはすべて削除されますので、テンポラリディレクトリにファイルを保存しないでください。

■ Edit メニュー

- **Undo**

操作を取り消して元の状態に戻します。Ctrl+z でも同じ動作をします。

(注) Undo は何度も実行することができますが、機能によっては Undo できないものもあります。
例えば、Object Customize Wizard による属性の編集は Undo することができません。

- **Redo**

Undo を取り消します。Shift+Ctrl+z でも同じ動作をします。Redo は Undo を行った直後にのみ有効です。

- **Cut**

選択されているノードを削除します。Ctrl+x でも同じ動作をします。ノードの選択は、2D/3D View ウィンドウ、または Scene Graph ウィンドウから行います。削除されたノードは、ペーストすることができます。

- **Copy**

選択されているノードをコピーします。Ctrl+c でも同じ動作をします。コピーされたノードは、ペーストすることができます。

- **Paste**

Cut または、Copy バッファ中のノードを選択されているノードにペーストします。ペーストできない場合（例えば、ノードの型が一致しない場合）には、警告メッセージが表示されます。

- **Delete**

選択されているノードを削除します。

- **Keyframe**


Interpolator ノードを編集するための Keyframe Editor ウィンドウを表示します。
詳しい操作方法については、「第5章 Keyframe Editor」(32 ページ)をご覧ください。

■ View メニュー

各ウィンドウを表示します。現在表示されているウィンドウのメニュー項目にはチェックマークが付きます。


■ Run メニュー

- **Run**


ワールドを実行し、動きを確認することができます。確認は、3D View ウィンドウで行います。  (Play ボタン) を押しても同じ動作をします。

(注) Run の状態では、Main ウィンドウ、2D/3D View ウィンドウ、Message ウィンドウ、Java Console ウィンドウのみ使うことができます。

- **Pause**

実行を一時停止します。  (Pause ボタン) を押しても同じ動作をします。もう一度 Pause ボタンを押すか、Play ボタンを押すと実行が再開されます。

- **Stop**

実行を中止し、編集モードに戻ります。  (Stop ボタン) を押しても同じ動作をします。

■ Script メニュー

- **Expert**

Script のひな形を作成するための、Script Expert ウィンドウを起動します。詳しい操作方法については「第 12 章 Script Expert ウィンドウ」(67 ページ)をご覧ください。

- **Build**

変更があったスクリプトファイルのみをコンパイルします。

- **Rebuild**

すべてのスクリプトファイルを再コンパイルします。

(注) Script メニューは JDK1.1.x がインストールされている場合にのみ使うことができます。JDK1.2.x には対応していません。

■ Options メニュー

- **Environment**

環境設定ダイアログボックスが表示されます。タブをクリックすることにより、以下の設定項目を切り替えることができます。

General ページ



図 3 General ページ

Blink bounding box when selected :

オブジェクトを選択しているとき、オブジェクトを囲むバウンディングボックスを点滅させるかどうかを設定します。

Show all nodes in Keyframe Editor source :

Keyframe Editor のソース選択コンボボックスに、指定できるすべてのノードを表示させるかどうかを設定します。

Grid ページ

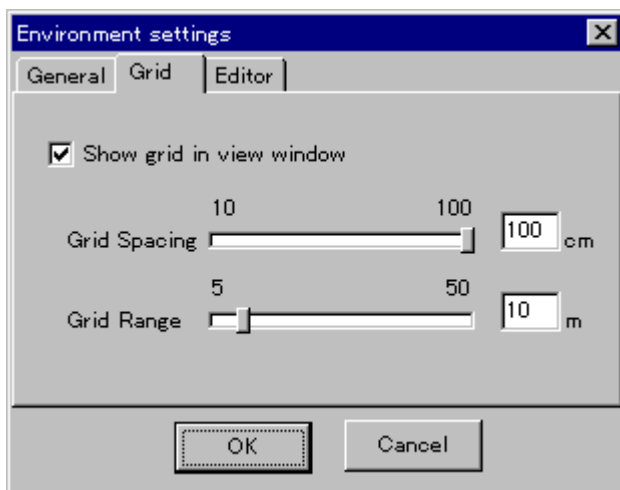


図 4 Grid ページ

Show grid in view window :

2D/3D View ウィンドウに、グリッドラインを表示させるかどうかを設定します。

表示させる場合は、グリッドの間隔と範囲をスライダーバーもしくは数値入力により、それぞれ設定することができます。

Editor ページ

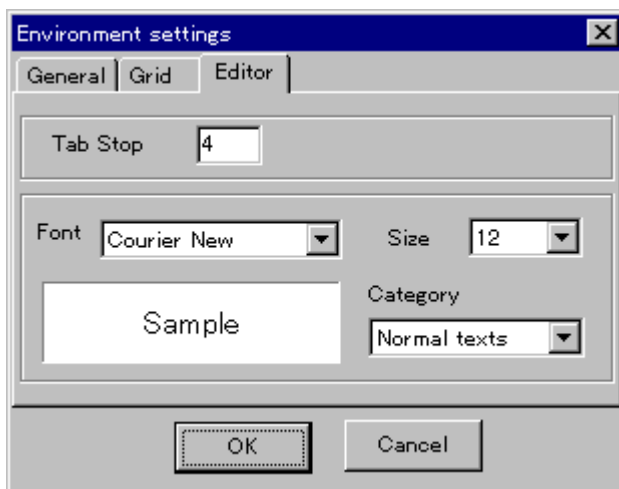


図 5 Editor ページ

Text Editor 内の Tab の文字数、書式設定として、Font、Size、カテゴリー（通常のテキスト、キーワード、コメント、背景色）毎の色を設定することができます。

■ Help メニュー

● VRML97 Spec

VRML97 関するスペックを HTML ブラウザー上に表示させます。

- （注1）表示を行うためには、Netscape Navigator 3.0 以降、または、Microsoft Internet Explorer 3.0 以降が必要です。
- （注2）VRML97 オンラインマニュアルの Copyright については、Conductor をインストールしたフォルダ内の doc\vrml97\copyright.html をお読みください。
- （注3）Netscape Navigator または、Microsoft Internet Explorer を起動していない状態で Help メニューを選択した場合には、ブラウザを起動するだけで、ヘルプは表示されません。ヘルプを表示するには、Help メニューを再度選択し直してください。

- **Show CLASSPATH**

Sun の JRE 1.1.x あるいは JDK 1.1.x がインストールされている場合に、クラスパス情報を表示します。

(注1) JRE/JDK が検出されない場合には、クラスパス情報は表示されません。

(注2) スクリプトをコンパイルする場合には JDK が必要です。JRE/JDK はバージョン 1.1.x をインストールしてください。JDK1.2.x には対応していません。

- **About**

図 6 に示す About ダイアログボックスを表示します。このダイアログボックスには、Conductor のバージョンや、Conductor が認識した Java に関する情報が表示されます。

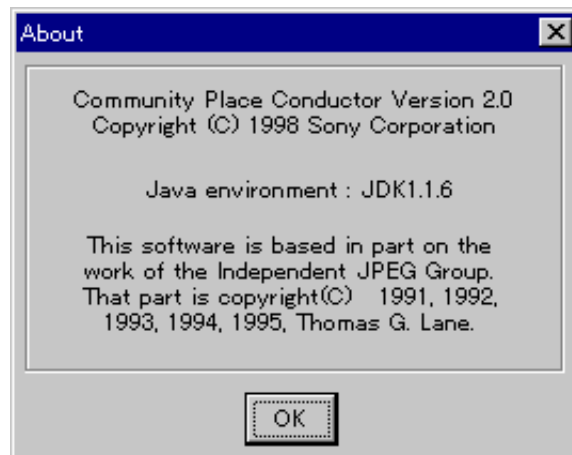






図 6 About ダイアログボックス

■ マウスモード

Conductor では以下のマウスモードボタンがあります。

表2 マウスモード

ボタン	機能
 Walk	2D/3D View ウィンドウにおける視点の移動
 Move	オブジェクトの移動
 Rotate	オブジェクトの回転
 Scale	オブジェクトのスケール変更

(注) マウスモードは Walk、Examine、Move、Rotate、Scale のいずれかの状態になります。Examine に関しては「Mouse Mode」(22 ページ)をご覧ください。

オブジェクトの移動、回転、スケールの変更は選択されているオブジェクトに対して行われます。オブジェクトの選択は、View ウィンドウ、または Scene Graph ウィンドウから行います。選択するとオブジェクトを囲むバウンディングボックスが表示されます。

■ ツールバー

ワールドにジオメトリやセンサーなどのノードを追加するには、図7のツールバーで追加したいノードのボタンを押し、3D View ウィンドウをクリックします。(実際の使い方については、チュートリアルを参照して下さい)



図7 ツールバー

表 3 Geometry






ボタン	ノード
 Box	立方体
 Cone	円錐
 Cylinder	円柱
 Sphere	球
 Text	文字
 IndexedFaceSet	多面体（Conductor では四角形のみ生成できます）
 ElevationGrid	地表の様な凹凸面（Conductor では平面のみ生成できます）
 Extrusion	平面形状の押し出しにより生成される立体（Conductor では直方体のみ生成できます）

表 4 Sensor







ボタン	ノード
 ProximitySensor	領域内に視点が含まれるか検出するセンサー
 TimeSensor	時間に関するセンサー
 TouchSensor	オブジェクトのヒット（マウスによるクリックなど）を検出するセンサー
 PlaneSensor	ドラッグした距離をローカル座標系におけるxy平面上の移動量として検出するセンサー
 SphereSensor	ドラッグした距離を回転量として検出するセンサー
 VisibilitySensor	領域が可視範囲に含まれるかを検出するセンサー

表 5 Common





ボタン	ノード
 DirectionalLight	平行光源
 PointLight	点光源
 SpotLight	スポットライト
 WorldInfo	ワールド情報

表 6 Bindable










ボタン	ノード
 Fog	フォグ
 Viewpoint	視点
 Background	背景
 NavigationInfo	ナビゲーション情報

表 7 Group

ボタン	ノード
 Anchor	URL へのリンク
 Billboard	視点に対して特定の方向を保つようにオブジェクトを自動的に回転させる
 Collsion	視点とオブジェクトの衝突を行うグループ
 Group	グループ
 Transform	ローカル座標系の定義
 Inline	他のファイルからのデータの読み込み

第2章 3D Perspective View ウィンドウ

図8に示す 3D Perspective View ウィンドウ (以下 3D View ウィンドウ) では、オブジェクトを 3D 透視投影のサーフェイスモデルで表示します。

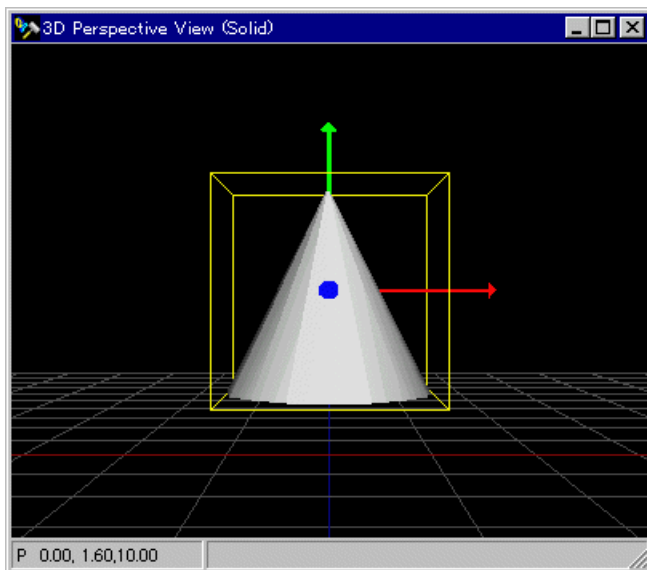


図8 3D View ウィンドウ

■ VRML ファイルの読み込み

3D View ウィンドウに VRML ファイルをドロップすることで、オブジェクトをシーンに追加することができます。

(注) 3D View ウィンドウにファイルをドロップする場合、いくつかの制限があります。詳しくはオンラインの制限事項 (Conductor をインストールしたフォルダ内の doc¥restrict¥restrict-j.txt) をご覧ください。

■ オブジェクトの選択

オブジェクトをマウスでクリックすると、選択されたオブジェクトを囲むバウンディングボックスと、そのオブジェクトのローカル座標軸が表示されます。X 軸が赤、Y 軸が緑、Z 軸が青で、矢印が各々の正方向を表します。

■ 視点位置と視線方向の表示

視点位置と方向は、左下隅のパネル（視点位置表示パネル）に数値で表示されます。

このパネルをクリックすることにより表 8 の様に表示を切り替えることができます。

表 8 視点位置表示パネルに表示されるパラメーター

表示内容	パラメーター	数値例
ワールド座標系での視点位置	(P x, y, z)	P 0.00, 0.00, 4.16
視点と選択ノード中心までの距離	(D d)	D 8.83
ワールド座標系での視線方向	(V x, y, z)	V -0.00, -0.00, -1.00

■ オブジェクトの生成

オブジェクトの生成は、Main ウィンドウのボタンによる生成や Library ウィンドウからのドラッグ&ドロップで行うことができます。

生成される位置は、マウスポインターの位置に関係無く常に、視点位置の前方に 10m 先の位置に生成されます。目の前に生成されるので、3D View ウィンドウの中心に表示されます。

● Main ウィンドウからのノードの生成方法

1. Main ウィンドウのツールバーから生成したいノードを選択する。
2. 生成したいノードのボタンをクリックする。
クリックしたボタンが黄色に変わり、押し込まれた状態になります。
3. マウスポインターを 3D View ウィンドウ上に移動させる。
マウスポインターの形状が変わります。
4. その状態でクリックする。
選択したノードが生成されます。

(注1) ノードによってその下にオブジェクトを追加できるものとできないものがあります。オブジェクトを生成する際には、その時選択されているノードを確認してください。

(注2) ジオメトリや光源など Transform と共に生成されるノードは 3D View ウィンドウに表示あるいは、反映されますが、センサーやスクリプトなど不可視のノードを生成しても 3D View ウィンドウには表示されません。Scene Graph ウィンドウには生成されたノードが表示されます。

■ナビゲーションの方法


3D View ウィンドウでの視点の移動は、Main ウィンドウでマウスモードを  Walk にして行います。マウスの左ボタンでドラッグするか、キーボードの矢印キーを使って、ナビゲーションします。


表9 ナビゲーション

動き	修飾キー	ドラッグ方向	キー	視点位置	視線方向
前方移動		上		移動	不変
後方移動		下		移動	不変
右へ回る		右		不変	移動
左へ回る		左		不変	移動
上方移動	Shift	上		移動(注1)	不変
下方移動	Shift	下		移動(注1)	不変
右方移動	Shift	右		移動(注2)	不変
左方移動	Shift	左		移動(注2)	不変
上を向く	Shift + Ctrl	上		不変	移動
下を向く	Shift + Ctrl	下		不変	移動

(注1) 視点位置の Y 座標のみ変化します。

(注2) 視点位置の Y 座標は不変で、X、Z 座標は、視線方向に垂直な面上をスライドします。

■ オブジェクトの移動


3D View ウィンドウ上でのオブジェクトの移動は、Main ウィンドウでマウスモードを  Move にして行います。

対象となるオブジェクトを選択し、オブジェクトを囲むバンディングボックスをドラッグすることで移動させることができます。

オブジェクトの移動は、ドラッグを開始する際に選択したバンディングボックスの面に沿った移動になります。

(注) View ウィンドウ上で、オブジェクトが遠方にある場合には、オブジェクトを囲むボックスを選択できないことがあります。


■ オブジェクトの回転

3D View ウィンドウ上でのオブジェクトの回転は、Main ウィンドウでマウスモードを  Rotate にして行います。

対象となるオブジェクトを選択し、バンディングボックスをドラッグすることで回転させることができます。

オブジェクトの回転は、オブジェクトの中心周りの回転になります。

■ オブジェクトのスケーリング

3D View 上でのオブジェクトのスケーリングには、Main ウィンドウでマウスモードを  Scale にして行います。

対象となるオブジェクトを選択し、バンディングボックスをドラッグすることでスケーリングすることができます。

オブジェクトのスケーリングは、オブジェクトの中心を原点としたものになります。

(注) マウスによるスケーリングは、等方スケーリングのみとなります。

■ ポップアップメニュー

図9に示すポップアップメニューは、3D View ウィンドウ上でマウスの右ボタンをクリックすると表示されます。

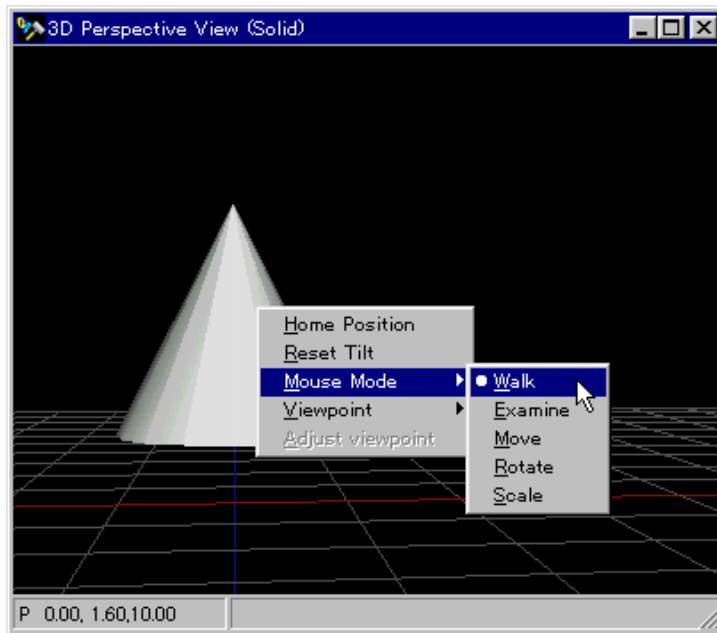


図9 3D View ウィンドウでのポップアップメニュー

ポップアップメニューは以下のサブメニューで構成されます。

● Home Position

自分がどこにいるかわからなくなった時や、開始位置に戻りたい場合に使用します。

(注) VRML ファイル内に Viewpoint ノードが存在する場合、最初の Viewpoint が Home Position となります。存在しない場合には、位置 (0, 1.6, 10) が HomePosition に対応します。

● Reset Tilt

視線方向を水平状態に戻します。

● Mouse Mode

マウスモード (Walk、Examine、Move、Rotate、Scale) を変更するためのメニューです。マウスモードはキー操作でも変更することができます。それぞれ、W、E、V、R、S キーに割り当てられています。

Examine

選択したオブジェクトを観察するときに使用します。3D View ウィンドウでのみ使用できます。

以下の手順に従って操作します。

1. ウィンドウでマウスモードを  Walk にする。
2. 観察したいオブジェクトを選択する。
3. ポップアップメニューの Mouse Mode Examine を選択する。

(注1) Examine できるオブジェクトが選択されている時のみ使用できます。
Examine モード中に、他のオブジェクトを選択することもできます。

Examine モードになり、選択したオブジェクトがウィンドウの中心に来るように視線方向が変わります。

(注2) 上下にドラッグしながら、シフトキーを押すとオブジェクトまでの距離が変化します。

4. オブジェクトの周りでマウスの左ボタンを押しながらドラッグする。
マウスの動きに応じて視点位置、視線方向が変わり、さまざまな角度からオブジェクトを観察することができます。

(注3) Examine モード中は、3D View では通常のナビゲーションはできません。
Parallel View では、ナビゲーションのみ可能です。

Examine を終了するには、再度ポップアップメニューで Examine もしくは他のマウスモードを選びます。

● Viewpoint

ワールドに存在するすべての Viewpoint ノードが表示されます。表示された Viewpoint を選択することにより、指定した場所に移動することができます。

(注) Viewpoint メニューには description で指定された文字列が表示されます。File New で作成した場合には Viewpoint は設定されていません。

- **Adjust Viewpoint**

選択されている Viewpoint を、現在の視点の位置と方向にあわせます。その結果、Viewpoint の position と orientation の値を調整することができます。このメニューは、Viewpoint が選択されている時にのみ使用できます。

第3章 Appearance ウィンドウ

Appearance ウィンドウでは、選択されているオブジェクトの色やテクスチャーの設定を行います。このウィンドウでは、Appearance ノードに含まれる Material ノード、ImageTexture ノード、TextureTransform ノードの編集を行うことができます。各ノードの使い方については、HelpメニューのVRML97 Specをご覧ください。

■ Material ノードの編集

material タブをクリックすると、図 10 に示す material ページが表示されます。ここでは、Material ノードの DEF 名と属性を設定することができます。

(注) DEF 名は DEF 名編集エリアに入力後、Enter キーを押すことで確定されます。

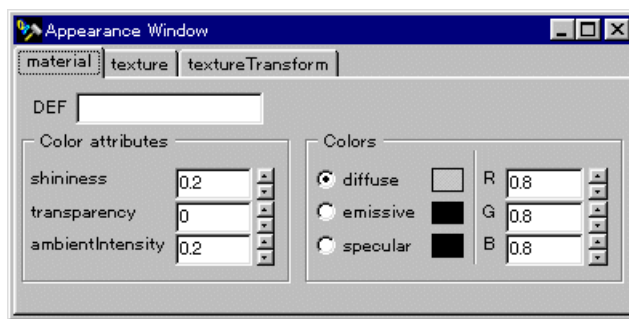


図 10 material タブが選択されている様子

● ColorMap ダイアログボックスを使った色設定

ダイアログボックスを使って各色を設定することができます。それぞれの色（diffuse、emissive、specular）の右隣にある長方形の部分をクリックすることで、図 11 に示す ColorMap ダイアログボックスが起動します。左側のカラーパネル上でマウスをドラッグすることにより、色相（H）と彩度（S）を、スライダーバーを動かすことにより明度（V）を設定することができます。

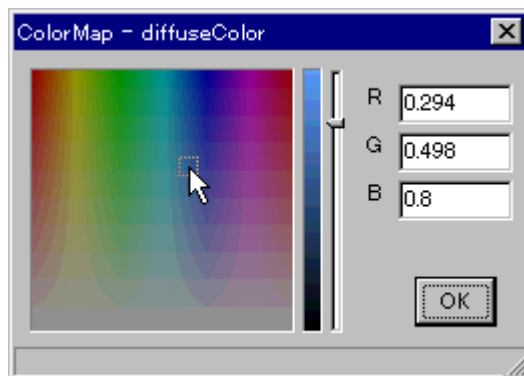


図 11 ColorMap ダイアログボックス

■ Texture ノードの編集

texture タブをクリックすることで図 12 に示す texture ページが表示されます。ここでは、ImageTexture ノードの DEF 名と属性を設定することができます。

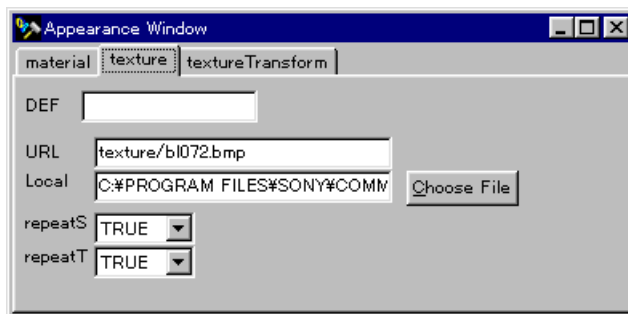


図 12 texture タブを選択した様子

（注 1）MovieTexture、PixelTexture ノードには対応していません。

(注2) DEF 名は DEF 名編集エリアに入力後、Enter キーを押すことで確定されます。

Local は実際のテクスチャファイルがある場所を示しています。通常は URL で指定した場所 (相対 URL 形式で記述された場合には、絶対 URL 形式に変換したもの) と Local は同じになりますが、Library ウィンドウからドロップした直後では、Library のディレクトリを指しています。

Save あるいは Save As した場合は URL で指定した場所にファイルがコピーされるため、URL と Local は同じファイルを参照することになります。Local は texture ページの Choose File ボタンを押すことで変更することができます。

■ TextureTransform ノードの編集

textureTransform タブをクリックすることで図 13 に示す textureTransform ページが表示されます。

ここでは、TextureTransform ノードの DEF 名、テクスチャをマッピングする際の移動、回転、スケールなどのパラメーターを設定します。

(注) DEF 名は DEF 名編集エリアに入力後、Enter キーを押すことで確定されます。

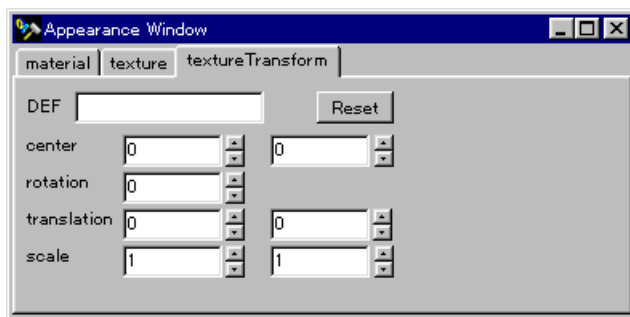


図 13 texture Transform タブを選択した様子

第4章 Attribute ウィンドウ

図 14 に示す Attribute ウィンドウでは、選択されているノードの属性と DEF 名を設定することができます。ウィンドウの下にあるステータスバーにはマウスポインター上にあるフィールドに関する、種類、変数の型、ノードの名前が表示されます。

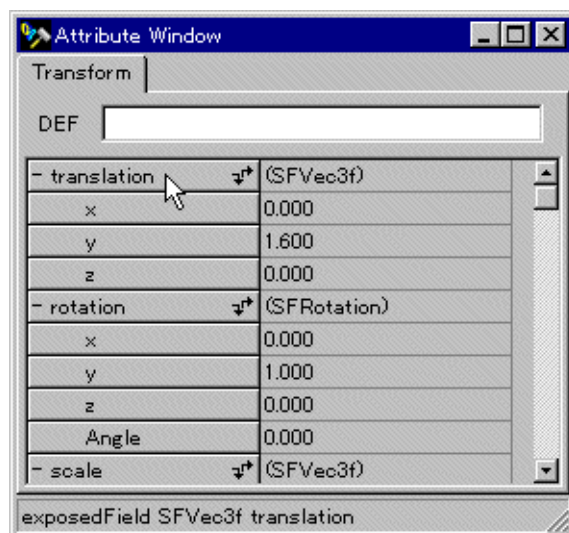


図 14 Attribute ウィンドウ

フィールド名の右側に表示されているアイコンは、表 10 に示すようにフィールドの種類を表しています。

表 10 アイコンによるフィールドの種類の表示

アイコン	フィールドの種類
	field
	exposedfield
	eventIn
	eventOut

(注) eventIn、eventOut は Script ノードと Proto インスタンスに対してのみ表示されます。

■ DEF 名の編集

DEF 名は、イベントをルーティングする際や USE を使ってノードを生成する際に使用します。

(注) DEF 名は DEF 名編集エリアに入力後、Enter キーを押すことで確定されます。

■ 属性の編集

属性値の型は、数値型、文字列型、Color 型、BOOL 型の 4 種類があり、VRML97 が規定する field の型に対応しています。

BOOL 型を編集するには、「TRUE」または「FALSE」と表示されているエリアをダブルクリックして値を変更します。

編集エリアに「(SFCOLOR)」と表示されているものは、Color 型です。そのエリアをダブルクリックすると、ColorMap ダイアログが表示されますので、色を設定し、OK ボタンを押してください。

それ以外の型のフィールドは編集エリアに値を入力し、Enter キーまたは上下矢印キーを押すとワールドに値が反映されます。

(注) 項目の左端に「+」または「-」の表示があるものは、編集エリアを隠したり表示させたりすることができます。「+」が表示されている項目をダブルクリックすると編集エリアが表示されます。「-」が表示されている項目をダブルクリックすると編集エリアを隠すことができます。

■ Wizard タブ

Attribute ウィンドウよりも簡単にオブジェクトの属性を編集できる Object Customize Wizard という機能も用意されています。選択しているオブジェクトが Wizard に対応している場合、Attribute ウィンドウに Wizard タブが表示され、わかりやすい GUI によって属性の編集をすることができます。Wizard ライブラリにあらかじめ用意されているライブラリオブジェクトは Wizard に対応していますので、このオブジェクトを 3D View ウィンドウにドラッグ & ドロップすると、図 15 に示す Wizard タブが表示されます。スライダーバーなどを操作して、好みに合わせてオブジェクトをカスタマイズすることができます。

(注1) Attribute ウィンドウでの変更は Wizard には反映されません。

(注2) Object Customize Wizard による属性の編集は Undo することができません。

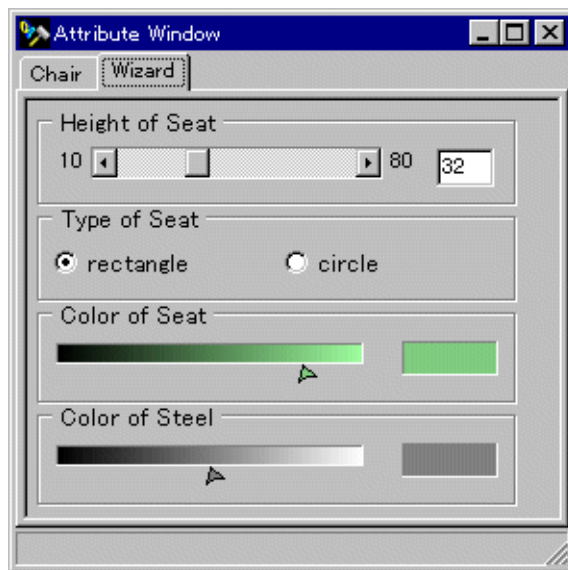


図 15 Object Customize Wizard

■PROTO 内のノード編集

PROTO 内のノードが選択された場合、Attribute ウィンドウには eventIn、eventOut を含むすべてのフィールドが表示されます。また、これらのフィールドは IS を使ったマッピングが可能です。(アンマップすることもできます。)

(注) PROTO 内のノード選択に関しては「第 11 章 Scene Graph ウィンドウ」(60 ページ)をご覧ください。

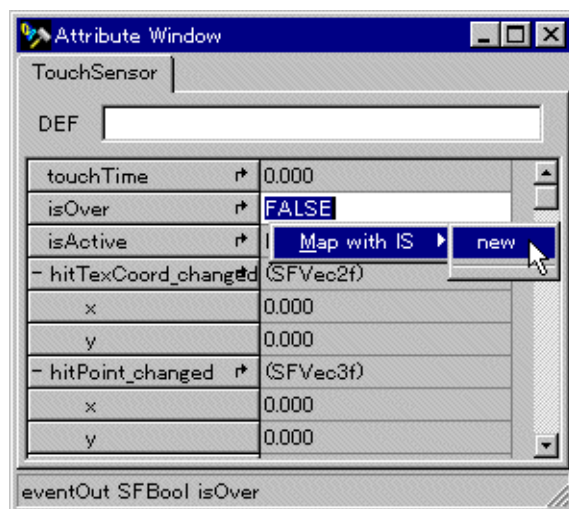


図 16 PROTO 内のノード (TouchSensor) が選択された場合

• ポップアップメニュー

map with IS new

IS を使って新規にフィールドをマッピングします。このときに図 17 に示すダイアログボックスが現れ、フィールド名を設定することができます。

新規に field または exposedField をマッピングする場合、マッピングする前のフィールドの値が、デフォルト値となります。

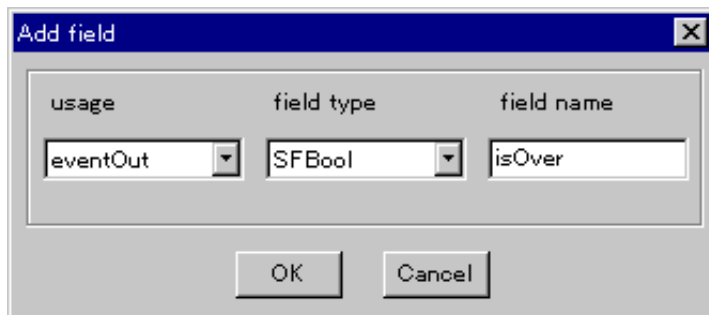


図 17 フィールドを追加するためのダイアログ

(注) PROTO 内の編集に関しては、オンラインの制限事項(Conductor をインストールしたフォルダ内の doc¥restrict¥restrict-j.txt)をお読みください。

unmap from IS

マッピングされているフィールドを解除します。

■ Script ノードに対する編集

Script ノードが選択されている場合、ポップアップメニューの Remove field、Add field によりフィールドの削除、追加を行うことができます。ただし、class ファイルがある場合は、ソースファイル (.java ファイル) が存在している必要があります。また、URL が指定されていない場合は、フィールドの追加、削除を行うことが可能です。

第5章 Keyframe Editor ウィンドウ

図 18 に示す Keyframe Editor ウィンドウでは、VRML97 で規定される Interpolator ノードを編集することができます。Keyframe Editor を起動すると Keyframe 編集モードになります。また、exposedField が変更されると自動的に対応する Interpolator ノードが生成され、さらにルーティングも自動的に追加されます。通常の編集モードに戻るには Keyframe Editor を終了させてください。

(注) Keyframe 編集モードではオブジェクトの編集は Interpolator に対する編集となり、オブジェクト自身の属性値は変更されません。オブジェクト自身の編集を行うには Keyframe 編集モードを終了させて下さい。

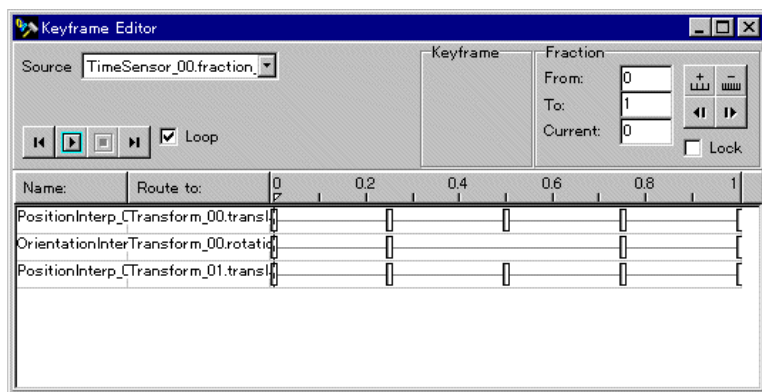




図 18 Keyframe Editor ウィンドウ

(注) CoordinateInterpolator、NormalInterpolator には対応していません。

■ Keyframe Editor の起動および終了方法



Keyframe Editor は Main ウィンドウの  を押すか、メニューから Edit Keyframe を選択します。



終了するには、Main ウィンドウの  をもう一度押すか、メニューから Edit Keyframe を選択します。また、ウィンドウ右上の×ボタンをクリックし、Keyframe Editor ウィンドウを閉じて終了させることもできます。

■ Source の選択

Source コンボボックスから Interpolator を制御するためのフィールド（もしくはイベント）を選択します。通常は、TimeSensor の fraction イベントとなります。ここに表示させるためには、あらかじめ TimeSensor などに DEF 名をつけておく必要があります。

■ 表示の拡大縮小、スクロール

（拡大ボタン）または （縮小ボタン）をクリックすることで、表示エリアを拡大したり、縮小することができます。

 または （スクロールボタン）をクリックすることで、表示エリアをスクロールさせることができます。

■ Fraction の設定

● From フィールドと To フィールド

Keyframe Editor が表示する範囲を指定します。これらの値は拡大、縮小ボタン、スクロールボタンを押した場合にも変化します。


● Current フィールド

キーフレームを追加する場所を指定します。この値は、Fraction パネル（目盛りがついている部分）をクリックした場合にも変化します。

■ 表示のロック

Lock チェックボックスをチェックすると、拡大、縮小ボタン、スクロールボタン、From フィールド、To フィールドが固定されます。

■ アニメーションの動作確認

Keyframe Editor 中の （Play ボタン）を押すことで、現在の Source に関連づけられた Interpolator の動作を確認することができます。

(注) ワールド全体の動作を確認するためには、Keyframe Editor を終了して、Main ウィンドウの Play ボタンを押してください。

■ Loop チェックボックス

アニメーションをループさせるかどうかを指定します。Source コンボボックスに TimeSensor が指定されている場合には、TimeSensor の loop フィールドと連動します。

■ キーフレームの複数選択

同一の Interpolator に属するキーフレームは、Shift キーを押しながらマウスをクリックすることで複数選択することができます。選択されたキーフレームは赤くなります。

■ ポップアップメニュー

キーフレーム編集エリアでマウスの右ボタンをクリックすると図 19 に示すポップアップメニューが表示されます。

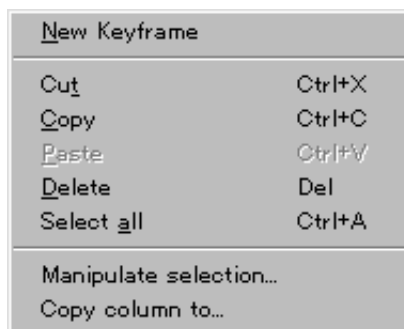


図 19 Popup メニュー

● New Keyframe

現在の Fraction で指定された位置に新しいキーフレームを追加します。

● Cut

選択されたキーフレームをカットします。現在選択されている Interpolator に対してのみ有効です。

- **Copy**

選択されたキーフレームをコピーします。現在選択されている Interpolator に対してのみ有効です。

- **Paste**

カットまたは、コピーした Keyframe をペーストします。

- **Select all**

選択された Interpolator に含まれるすべてのキーフレームを選択します。

- **Manipulate selection**

選択された複数のキーフレームに対して、以下で説明するキーフレームの操作を行います。

- **Copy column to**

現在の Fraction に含まれる（全ての Interpolator の）キーフレームを他の Fraction にコピーします。この操作は、複数の Interpolator に対して有効です。

■ キーフレームの操作（スケーリング、スムージング）

選択されている複数のキーフレームに対して、スケーリングしたりスムージングすることができます。複数のキーフレームを選択した状態で、マウスの右ボタンを押して図 19 に示すポップアップメニューから「Manipulate selection」を選んでください。

（注）スムージングは PositionInterpolator に対してのみ有効です。

- **スケーリング**

図 20 に示す Scale タブを選択すると、スケールページが表示されます。ここでは、キーフレームをスケーリングすることができます。

Scaling Factor :

スケーリングの値を指定します。

50% : 倍速

200% : 半分の速度

100% : 反転

Custom: 数値（パーセント）で指定

Scaling point :

スケーリングの中心点を指定します。

Left-hand keyframe : 左端のキーフレーム

Center of selection : 選択されているキーフレームの中心

Right-hand keyframe : 右端のキーフレーム

Custom : 数値（目盛り）で指定

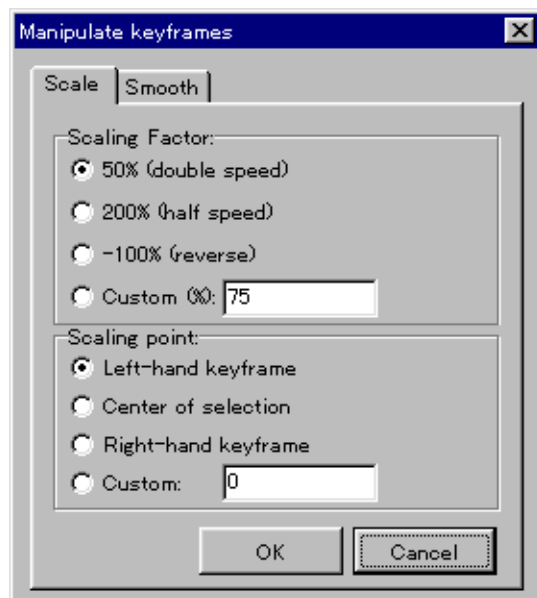


図 20 キーフレーム操作ダイアログ（Scale タブ）

• スムージング

図 21 に示す Smooth タブを選択すると、スムージングページが表示されます。ここでは、ベジェ曲線を使って中間のキーフレームを生成します。これにより、よりスムーズな補間が可能になります。

of smoothing keyframes:

既存のキーフレームに対して新規に挿入するキーフレームの数を指定します。

Tension、Bias:

ベジェ曲線のパラメーターを指定します。通常は、-1.0 から +1.0 の値を指定します。Tension が 1.0 の場合には、線形補間と同じになります。

Loop first and last keyframe

最初と最後のキーフレームの値が一致している場合に使用します。

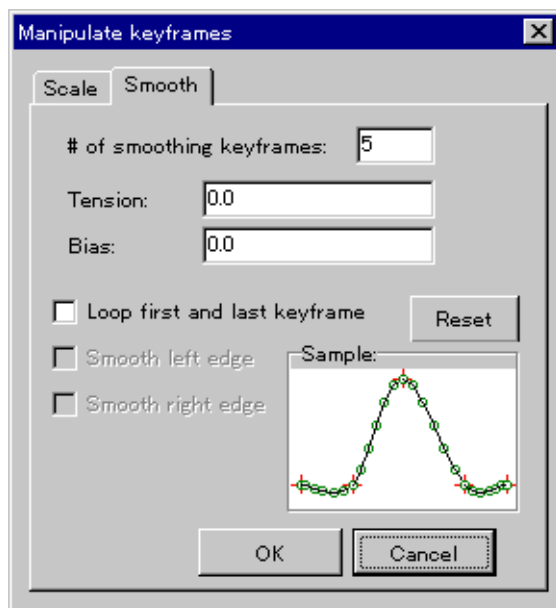


図 21 キーフレーム操作ダイアログ (Smooth タブ)

第6章 Library ウィンドウ

図 22 に示す Library ウィンドウには用意された各種のライブラリが表示されます。これらは、ドラッグ&ドロップによりワールドに追加することができます。それぞれのライブラリページはタブをクリックして切り替えます。

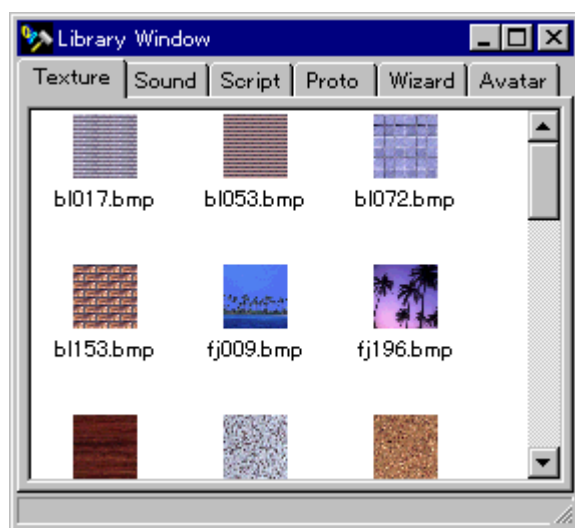


図 22 Library ウィンドウ

■ ドラッグ&ドロップ操作

● Texture

図 23 に示すように、2D/3D View ウィンドウ上のオブジェクトに対してドラッグ&ドロップすることでテクスチャを貼り付けることができます。

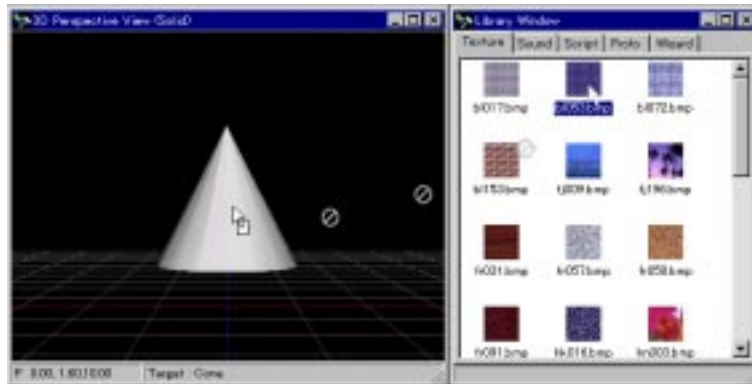


図 23 ドラッグ&ドロップによるテクスチャの貼り付け

● Sound、Script、Proto、Wizard

2D/3D View ウィンドウにドラッグ&ドロップして使います。

(注) PROTO 定義部もしくは Wizard 定義部に、PROTO ライブラリや Wizard ライブラリをドラッグ&ドロップすることはできません。

● Avatar

Multi User ウィンドウ、2D/3D View ウィンドウにドラッグ&ドロップして使います。

(注) Avatar ページはマルチユーザーワールドを編集のみ表示されます。詳しくは「第8章 Multi User ウィンドウ」(44 ページ)をご覧ください。

■ 関連アプリケーションの起動

アイコンをダブルクリックすることにより、関連づけられたアプリケーションを起動し、アイコンに対応するファイルの内容をプレビューできます。

■ Script ライブラリのヘルプ表示

Script ライブラリでは、アイコンを選択し、F1 キーを押すことにより各アイコンを解説するヘルプを表示することができます。ヘルプは、HTML ブラウザーに表示されます。

(注1) この機能を利用するためには、Netscape Navigator3.0 以降または、Microsoft Internet Explorer 3.0 以降が必要です。

(注2) Netscape Navigator または、Microsoft Internet Explorer を起動していない状態でヘルプメニュー、もしくは F1 キーを選択した場合には、ブラウザを起動するだけで、対象となるページにアクセスしません。Help を表示するには、Help メニュー、もしくは F1 キーを再度選択し直してください。

■ ライブラリフォルダーの変更

Livrar ウィンドウ内でマウスの右ボタンをクリックすると、現在表示中のライブラリが参照しているフォルダーを変更するためのポップアップメニュー (Set directory for this library.) が表示され、選択すると図 24 に示すフォルダー選択ダイアログボックスが開きます。

フォルダーを変更するためには、希望するフォルダーをクリックして開いた状態にしておき、OK ボタンを押します。



図 24 フォルダ選択ダイアログボックス

■ アイコンのカスタマイズ

Library ウィンドウでのアイコンは以下のルールにしたがって表示されます。

1. Texture ライブラリでは、テクスチャファイルの画像がそのままアイコンとして表示されます。
2. それ以外のライブラリでは、ライブラリファイルと同じディレクトリにある同名で拡張子が .bmp のファイルをアイコンとして使用します。もし、これが存在しない場合には、ライブラリディレクトリにある _default.bmp が使われます。このファイル名の先頭には _ (アンダーバー) が付きますので注意してください。
3. もし、_default.bmp も存在しない場合には、Conductor のアイコンが使用されます。
4. 各ライブラリファイルに対応した bmp ファイルを用意することで、アイコンをカスタマイズすることができます。

(注) アイコンファイルの大きさは32ドット×32ドット以上でなければなりません。

■ 各ライブラリでの対応フォーマット

● Texture

bmp、gif、jpeg が使用できます。

● Sound

wav、mod が使用できます。

(注) mod を使うためには、DirectX3.0 以降がインストールされている必要があります。

● Script

Script ノードを記述した VRML ファイルと、class ファイルが必要です。

● Proto

PROTO が記述された VRML ファイルです。VRML ファイルには、PROTO 定義と PROTO インスタンスが記述されている必要があります。

(注) PROTO インスタンスは、他のノードで囲まれてはいけません。

● Wizard

Object Customize Wizard に対応し VRML ファイルです。

Wizard に対応したオブジェクトを作成する方法は、「第 15 章 Object Customize Wizard の作成方法」(85 ページ) をご覧ください。

- **Avatar**

Avatar を記述した VRML ファイルです。

■ ライブラリへの追加

ライブラリフォルダーにファイルを追加もしくは削除することで、ライブラリオブジェクトを追加もしくは削除することができます。Texture ライブラリへ追加するときは、対応フォーマットの画像ファイルを Library ウィンドウにドラッグ&ドロップします。

Proto ライブラリへ Proto を追加するときは、Scene Graph ウィンドウのスコープを追加したい Proto に切り替え、ポップアップメニューから Register to the Library を選択します。

ライブラリオブジェクトの追加または削除の際、アイコンの表示は自動的に更新されます。

第7章 Message ウィンドウ

図 25 に示す Message ウィンドウでは実行時に発生した警告メッセージ、Java コンパイラからのエラーメッセージなどを表示します。

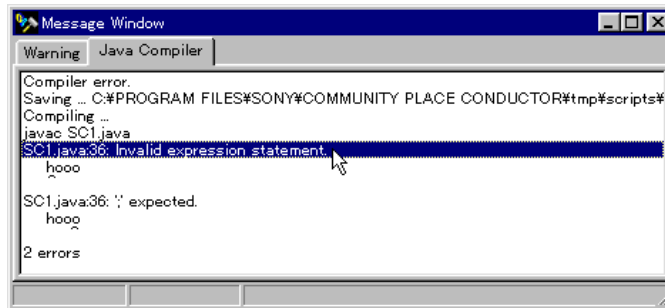


図 25 Message ウィンドウ

■ Warning メッセージ

Warning タブを選択することで、警告メッセージを見ることができます。

■ Java コンパイラからのメッセージ

コンパイル時のエラーメッセージなどが出力されます。コンパイルエラーが発生した場合には、Message ウィンドウのエラー行をダブルクリックすることで、Text Editor ウィンドウのエラーのある場所にジャンプすることができます。

第8章 Multi User ウィンドウ

Multi User ウィンドウでは、マルチユーザーワールドの作成時に、マルチユーザーサーバー（Community Place Bureau（以下 Bureau））の設定と、そのワールドで使用可能となるアバターの指定を行います。Bureau の使い方については、Bureau のマニュアルをご覧ください。

■ マルチユーザーワールドの作成

シングルユーザーワールドを読み込んだ場合は、Multi User ウィンドウには「Create Multi User world」と表示されたボタンが表示されます。現在編集中のコンテンツをマルチユーザーワールドにするためにはこのボタンを押してください。その結果、Sony_WorldInfo ノードが生成されマルチユーザーワールドになります。Scene Graph ウィンドウで、Sony_WorldInfo ノードを生成した場合もマルチユーザーワールドにすることができます。その後、以下のようにマルチユーザーサーバーの設定と、そのワールドで使用可能となるアバターの指定を行います。シングルユーザーワールドに切り替えるためには、Scene Graph ウィンドウで Sony_WorldInfo ノードを削除してください。

■ マルチユーザーサーバーの設定

Server タブをクリックすると、図 26 に示す Server ページが表示されます。ここでは、このコンテンツが接続するマルチユーザーサーバー（Bureau）のホスト名とポート番号を設定することができます。このページでの設定は Sony_WorldInfo ノードの cpBureau フィールドに反映されます。host 欄が空白の場合は、自動的に「localhost」が設定されます。作成したマルチユーザーワールドのテストのために、Bureau とコンテンツをローカルマシンで起動する場合はこのままで動作します。port 欄が空白の場合は、自動的に「5126」が設定されます。これは Bureau のデフォルトのポート番号です。

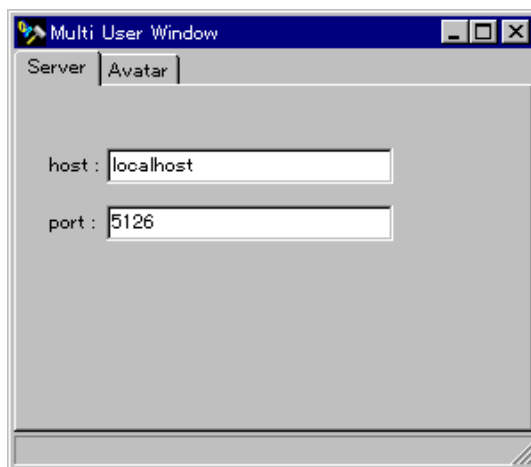


図 26 Server タブが選択されている様子

■ 使用アバターの指定

Avatar タブをクリックすることで図 27 に示す Avatar ページが表示されます。Avatar ページでは現在作成しているマルチユーザーワールドで、ユーザーが使用できるアバターのリストを表示します。

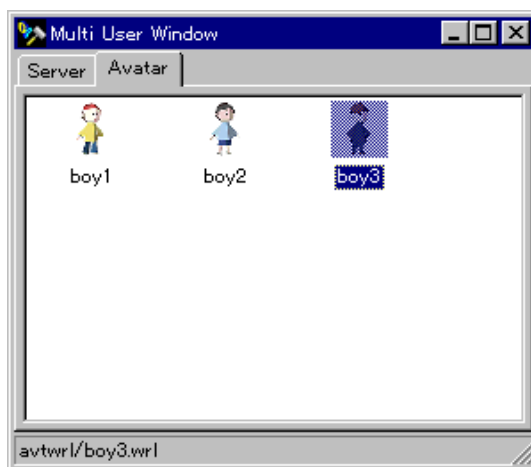


図 27 Avatar タブを選択した様子

- アバターのリストへの追加

使用できるアバターを追加したい場合は、Library ウィンドウの Avatar ページから Multi User ウィンドウの Avatar ページ（または 3D View ウィンドウ）へ、アバターのアイコンをドラッグ&ドロップします。Library ウィンドウの Avatar ページはコンテンツがマルチユーザーワールドの時のみ選択可能です。

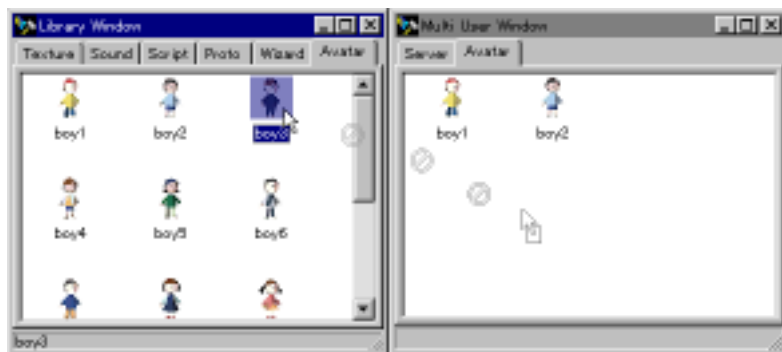


図 28 Library ウィンドウから Avatar ウィンドウへアバターをドラッグ&ドロップしているところ

- アバター名の変更

Multi User ウィンドウの Avatar ページでアバターアイコンを選択すると、保存時のアバターファイルの相対 URL がウィンドウ下のステータスバーに表示されます。相対 URL が同じアバターは追加することができません。この場合はアイコンの名前を変更して保存時のアバターのファイル名を変更してください。（ただし日本語は使用できません。）

例えば、ライブラリーの同じアバターを基に編集を加えてオリジナルのアバターを複数作成する場合や、読み込んだマルチユーザーワールドに含まれるアバターと、ライブラリーから追加しようとしたアバターが同じ URL となる場合には名前の変更が必要です。

- アバターのリストからの削除

使用できるアバターのリストから、あるアバターを削除するには Multi User ウィンドウの Avatar ページで、削除したいアバターを選択し、Delete キーを押すか、マウスの右ボタンをクリックして、ポップアップメニューから「Delete」を選びます。

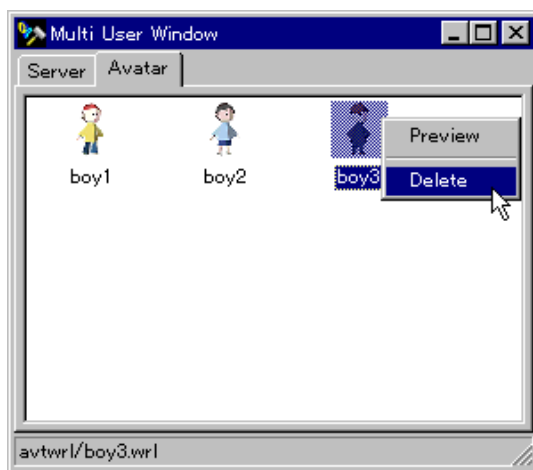


図 29 アバターの削除

- アバターの編集

使用できるアバターのリストに含まれるあるアバターを、確認および編集するには Multi User ウィンドウの Avatar ページで、編集したいアバターを選択し、マウスの右ボタンをクリックして、ポップアップメニューから「Preview」を選びます（または改行キーを押すか、マウスの左ボタンをクリック）。

すると、アイコンに赤い枠が表示され、3DView にアバターが表示されます。

Scene Graph にはアバターファイルが Inline ノードとして生成され、その Inline ノードのスコープに入ることによって、アバターの編集をすることができます。

ただしアバターはワールドに含まれるオブジェクトではなく、編集のために一時的にワールドに追加されただけであることに注意してください。ワールドの保存時にはアバターの Inline ノードはワールドから削除されます。

アバターの編集を終了するには、Avatar ページでそのアバターを選択し、再びポップアップメニューから「Preview」を選びます（または改行キーを押すか、マウスの左ボタンをクリックします）。

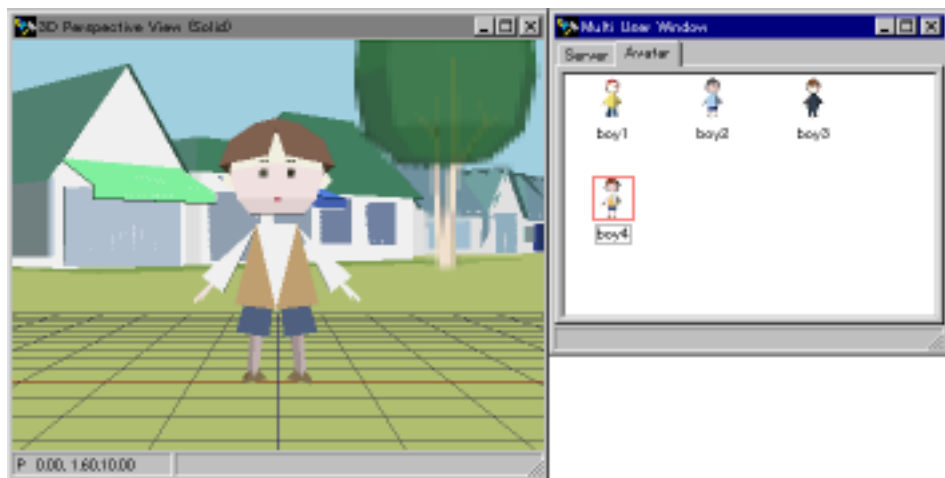
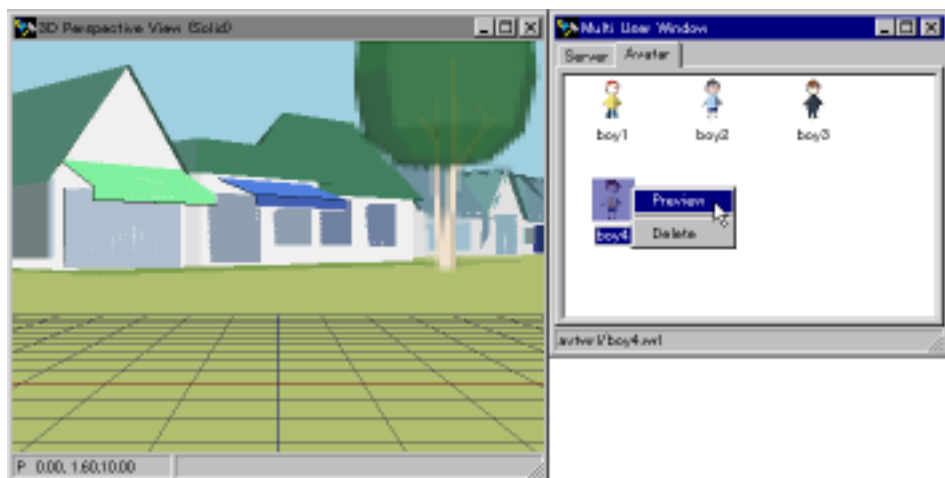


図 30 アバターを表示する前（上図）と表示したところ（下図）

■ マルチユーザーワールドの動作確認

作成したマルチユーザーワールドの動作確認をするには、まず Server ページで設定したホストとポートに従って Bureau を起動します。

その後編集集中のコンテンツを保存して Play ボタンを押すと、Conductor で動作の確認ができます。

アバターの姿や動作を確認するには、複数の Browser（または Conductor）を起動してそのコンテンツを読み込んでください。

第9章 Parallel View ウィンドウ

図 31 に示す Parallel View ウィンドウでは、3 つの Parallel View(平行投影の View) を表示します。3 つの Parallel View とは、上面図(Top View)、側面図(Side View)、正面図(Front View)です。これらの View ではオブジェクトは常にワイヤフレームモデルで表示されます。

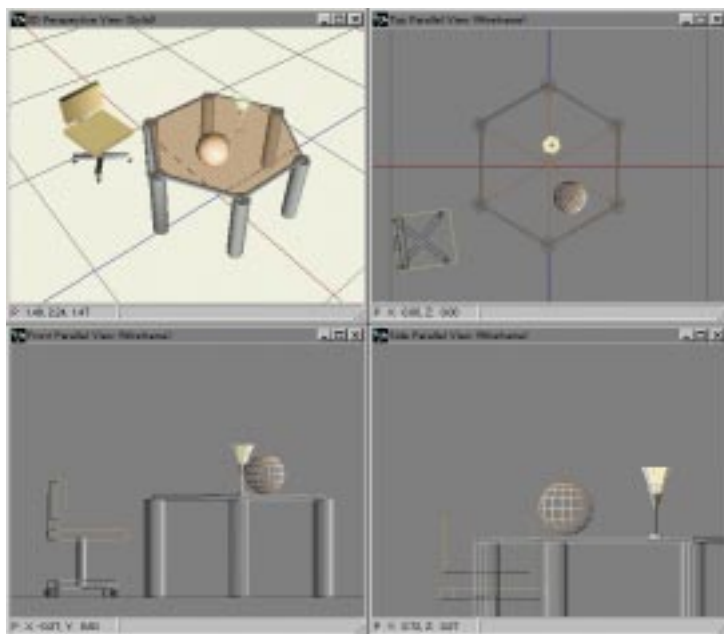


図 31 Parallel View ウィンドウ

■ 視点位置と視線方向の表示

視点位置と方向は、左下隅の視点位置表示パネルに数値で表示されます。

このパネルをクリックすることにより、表 11 に示すように表示を切り替えることが可能です。

表 11 視点位置表示パネルに表示される数値

表示内容	変数	表示例
ワールド座標系での視点位置	(P)	P Y: -0.47, Z: -2.13
現在のズーム率の割合	(S)	S 2.222
ワールド座標系での視線方向	(V)	V -1.00, -0.00, -0.00

Parallel View での視点位置、視線方向はワールド座標系で以下の表のようになります。

表 12 Parallel View の視点位置と視線方向

View の種類	視点位置	視線方向
Top View	X-Z 平面と平行な面上	-Y 軸固定
Side View	Y-Z 平面と平行な面上	-X 軸固定
Front View	X-Y 平面と平行な面上	-Z 軸固定

■ ポップアップメニュー

マウスの右ボタンをクリックすることにより表示されます。
ポップアップメニューは、以下のサブメニューで構成されます。

● Home Position コマンド

視点位置、視線方向、ズーム率をデフォルト値に戻します。3D View の場合と異なり、移動途中の様子は表示されません。Home Position の値は、固定で以下の表のようになります。

表 13 Home Position の値

view の種類	視点位置	視線方向	ズーム率
Top View	X = 0、Z = 0	-Y 軸固定	1.0
Side View	Y = 0、Z = 0	-X 軸固定	1.0
Front View	X = 0、Y = 0	-Z 軸固定	1.0

Mouse Mode、View Point に関する説明は 22 ページを参照してください。

■ オブジェクトの生成

3D View でのオブジェクトの生成と同様に Parallel View 内でもオブジェクトを生成することができます。Parallel View では、生成する際にマウスを押した地点の真下に生成されます。

また、生成されるオブジェクトのワールド座標は、各視線方向のワールド座標が 0 となる位置に生成されます。

例えば TopView でオブジェクトの生成を行った場合、オブジェクトはマウスポインターの位置に対応するワールド座標の位置 (X,0,Z) に生成されます。

■ ナビゲーションの方法

基本的には 3D View でのナビゲーションと同じです。

ただし、キーボードの矢印キーは使用できません。

各 Parallel View では、各々の視線方向に垂直な面上の移動のみ可能です。平行投影では視線方向で移動しても表示は変わりません。

以下の表に Parallel View でのナビゲーションの方法を示します。

表 14 ナビゲーションの方法

動き	修飾キー	ドラッグ方向	視点位置移動方向			ズーム率
			Top	Side	Front	
下方移動		下	+Z	-Y	-Y	
上方移動		上	-Z	+Y	+Y	
左方移動		左	-X	-Z	-X	
右方移動		右	+X	+Z	+X	
ズームイン	Shift	上				大
ズームアウト	Shift	下				小

Parallel View ウィンドウ内で Ctrl キーを押しながらクリックをすると、視点をその位置まで移動させることができます。

(注) ズーム率は初期値が1で、初期値に対する割合として増減します。最小で0.01倍、最大で16倍に制限されています。

■ オブジェクトの移動

操作方法は、3D View と同じで、視線に垂直な面内でオブジェクトを移動することができます。例えば、TopView でのオブジェクトの移動は X-Z 平面内で行われ、Y 軸方向の位置は変化しません。

■ オブジェクトの回転

操作方法は、3D View と同じですが、回転の方法に制限があります。Parallel View でのオブジェクトの回転軸は、そのオブジェクトのローカル座標原点を通り、その View での視線方向と平行な方向を持つ軸になります。




■ オブジェクトのスケーリング

操作方法は、3D View と同じです。ただし、スケール中心は 3D View のようにオブジェクトを囲むバウンディングボックスの中心でなく、オブジェクトのローカル座標の原点となります。このローカル原点の中心に向かってマウスをドラッグさせると縮小し、遠ざけると拡大します。

第10章 Route ウィンドウ

図 32 に示す Route ウィンドウでは、カレントスコープ内で発生するイベントのルーティングの表示および追加・削除を行います。カレントスコープについては、「第11章 Scene Graph ウィンドウ」(60 ページ)をご覧ください。

ウィンドウ内の左下のボタンをクリックすることで、以下の3つの表示に切り替えることができます。

-  : リスト表示
-  : リストとビジュアルの混合表示
-  : ビジュアル表示

■ リスト表示での操作

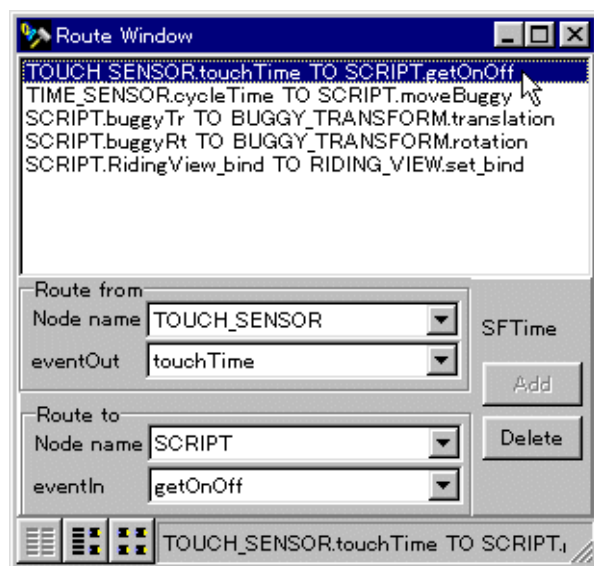


図 32 Route ウィンドウ

Route from

NodeName : イベントを出力するノードの名前

eventOut : イベントアウト名 (eventOut 型のフィールド名)

Route to

NodeName : イベントを受信するノードの名前

eventIn : イベントイン名 (eventIn 型のフィールド名)

(注) ノードにはあらかじめノード名が定義されている必要があります。

• ルートの追加

ルートの追加は以下のように行います。

1. イベントを出力するノードのノード名を NodeName リストから選択する。
2. eventOut をリストから選択する。
3. イベントを受けるノード名を NodeName リストから選択する。
4. eventIn をリストから選択する。

(注) リストに表示される eventIn は、eventOut の型と同じ型のものです。

5. Add ボタンを押す。
ルーティングが追加され、ルーティングリストに表示されます。

• ルートの削除

削除したいルートを選択し、Delete ボタンを押します。

■ ビジュアル表示での操作

ビジュアル表示モードでは、アイコンを使ってルートが表示行われます。アイコンはノード単位で表示され、ルートの追加、削除に応じて自動的に再レイアウトされます。また、DEF 名がついていてかつルートを持っていないノードは、ウィンドウ内の下部に表示されます。

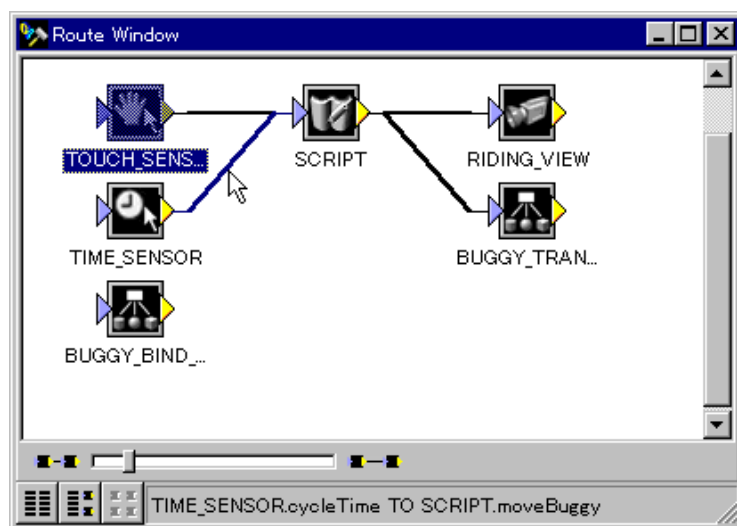


図 33 Route ウィンドウをビジュアル表示にした場合

● ルートの選択

ノードを結んでいる線をクリックすると、そのルートが選択され、Route ウィンドウの下部にあるステータスバーにルートの内容が表示されます。複数のルートがある場合にはポップアップで表示されます。いずれかのルートをクリックすることによって選択できます。

```
SCRIPT.buggyTr TO BUGGY_TRANSFORM.translation
SCRIPT.buggyRt TO BUGGY_TRANSFORM.rotation
```

図 34 複数のルートがある場合

• ルートの追加

Route ウィンドウ中のアイコンを他のアイコンにドラッグ&ドロップするか、Scene Graph ウィンドウとの間でドラッグ&ドロップします。Scene Graph ウィンドウのノードには DEF 名がついている必要はありません。(自動的に DEF 名が付きます)

詳しい方法については、チュートリアルをご覧ください。

• ルートの削除

ルートを選択しておき、右クリックでポップアップメニューを表示させ、削除したいルートを選択します。

```
Delete SCRIPT.buggyTr TO BUGGY_TRANSFORM.translation
Delete SCRIPT.buggyRt TO BUGGY_TRANSFORM.rotation
```

図 35 ルートを削除する時のポップアップメニュー

• アイコン間隔の調整

左下のスライダーバーにより、アイコン間隔を広げたり、狭めたりすることができます。ルートの状況が複雑な場合、見やすいように調整します。

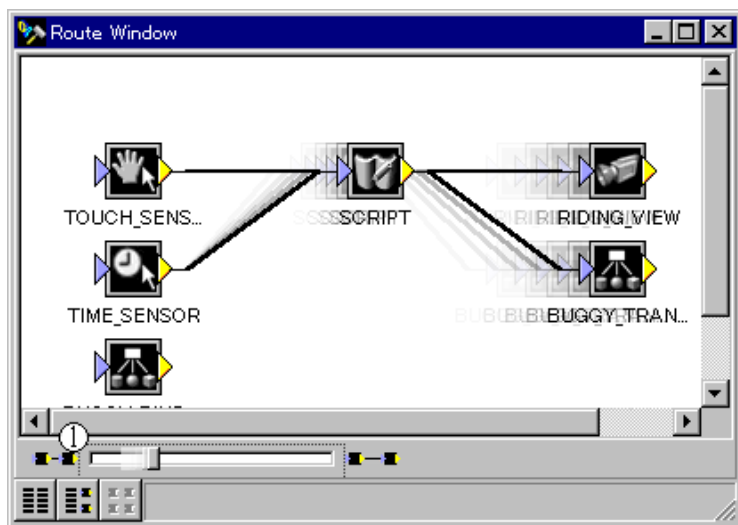


図 36 アイコン間隔を広げている様子

■ DEF 名の自動生成と変更

DEF 名のついていないノードを Scene Graph ウィンドウから Route ウィンドウへドロップすると、DEF 名が自動的に付加され、DEF 名を変更できる状態になります。DEF 名はノードの種類（例えば、TouchSensor の場合は TouchSensor_00）をもとに生成されます。

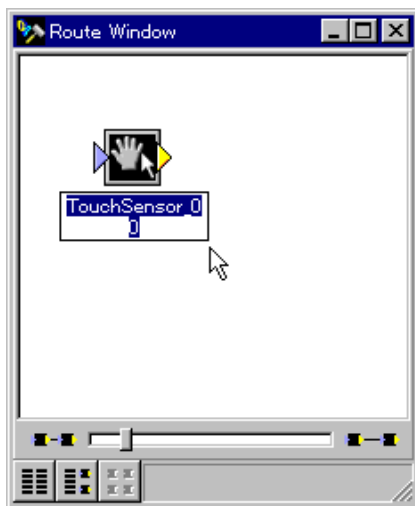


図 37 DEF 名が自動生成された直後

■ ポップアップメニュー

マウスの右ボタンでアイコンをクリックした場合：

そのアイコンのルーティング状態を確認することができます。

マウスの右ボタンでアイコン以外の部分ををクリックした場合：

● 2D Layout

ルート表示を 2 次元でレイアウトします。

● 1D Topological layout

ルート表示を 1 次元（接続の順番）でレイアウトします。

- **1D Alphabetical layout**

ルーティング表示を1次元（アルファベット順）でレイアウトします。

- **Animate changes**

この項目がチェックされていると、再レイアウトの様子をアニメーションで表示します。

- **Fade descendants**

Scene Graph ウィンドウ上で現在選択されているノードに関連するルートを破線で表示します。これにより、ノードを削除した場合にどのルートが削除されるのかを知ることができます。

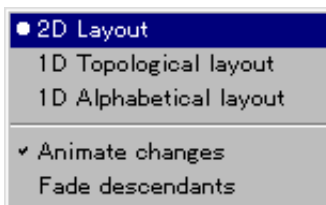


図 38 マウスの右ボタンでアイコン以外の部分を
クリックした時のポップアップメニュー

第 11 章 Scene Graph ウィンドウ

図 39 に示す Scene Graph ウィンドウは、カレントスコープのワールドのツリー構造を VRML97 のノード単位で表示します。また、シーングラフ上の任意のノードを選択したり、ドラッグ&ドロップによりノードを移動させたりすることも可能です。

ノードに DEF 名がついている場合には、□ で囲まれた DEF 名がノード名の後ろに表示されます。また、Inline ノードについては、URL 情報が <> の中に表示されます。

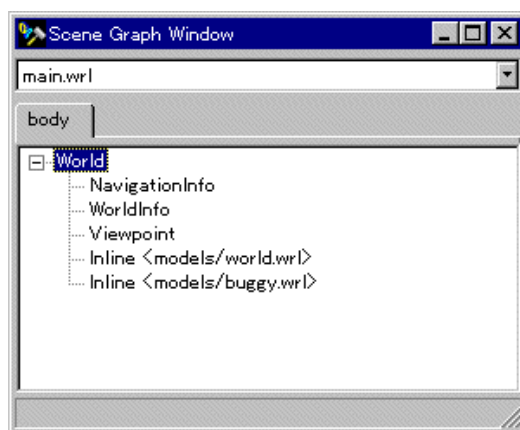


図 39 Scene Graph ウィンドウ

■ ファイルスコープの切り替え

VRML ファイルに Inline ノードが含まれる場合、それぞれに 1 つのスコープが与えられます。各ノードの編集は現在のスコープに対してのみ可能です。画面最上部のコンボボックスにより、編集対象となる「カレントスコープ」を切り替えることができます。

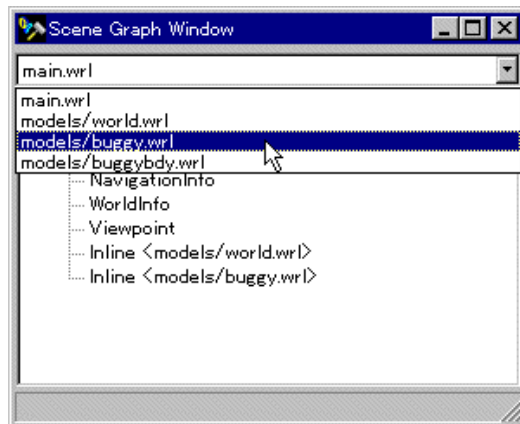


図 40 スコープ切り替えコンボボックス

■PROTO スコープの切り替え

現在選択されているファイル内に PROTO が含まれる場合、PROTO のスコープを切り替えるタブが表示されます。

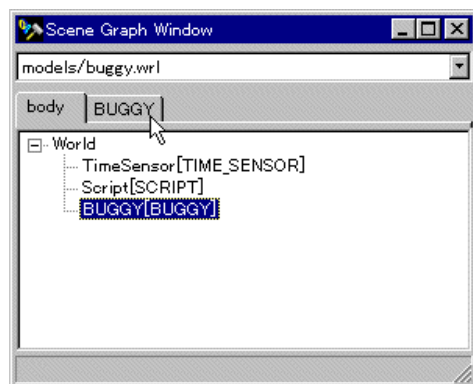


図 41 PROTO のスコープを切り替えるタブ

■ ノードの選択

選択したいノードをクリックします。選択されたノードは、Attribute ウィンドウで属性を変更できるようになります。ノードをダブルクリックすると、ノードが選択されると同時に、Attribute ウィンドウが表示されます。

■ ヘルプの表示

F1 キーを押すと HTML ブラウザーに VRML97 の解説ページが表示されます。オブジェクトが選択されていれば、そのノードの説明が表示されます。

(注) この機能は、Netscape Navigator 3.0 以降または、Microsoft Internet Explorer 3.0 以降にのみ対応しております。Netscape Navigator または、Microsoft Internet Explorer を起動してから F1 キーを押してください。

■ ポップアップメニュー

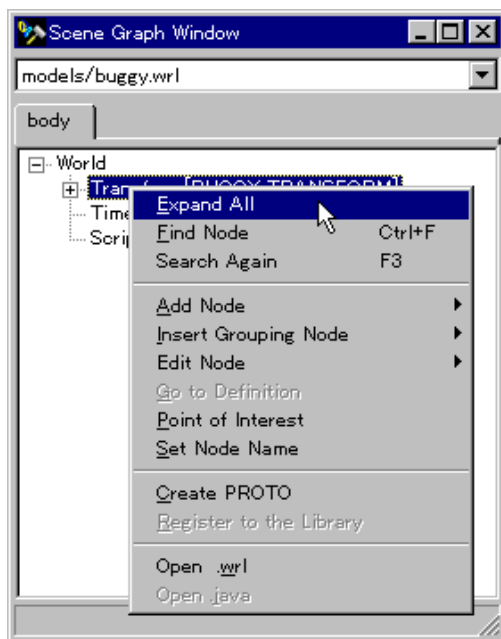


図 42 Scene Graph ウィンドウのポップアップメニュー

- **Expand All**

現在選択されているノードより下層にあるツリーを全て開きます。

- **Find Node**

図 43 に示すダイアログボックスが表示されます。

ノードの DEF 名で検索するには By Name タブを、ノードタイプで検索するには By Type タブをクリックすることで、それぞれのページを表示させます。

検索したい文字列はコンボボックスにより選ぶことができます。

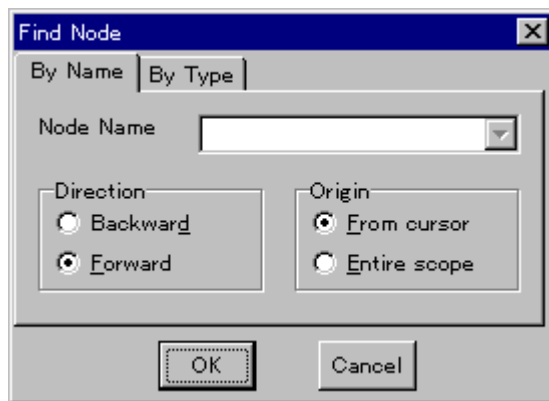


図 43 Find Node ダイアログボックス

Direction

Backward: 逆方向へ検索します。

Forward: 順方向へ検索します。

Origin

From cursor: カーソルのある位置から検索します。

Entire scope: ファイルの頭から検索します。

- **Search Again**

Find Node で設定した条件の下で再検索します。

- **Add Node**

VRML97 で規定されるノードを追加します。ノードはカテゴリー毎にサブメニューに分かれています。

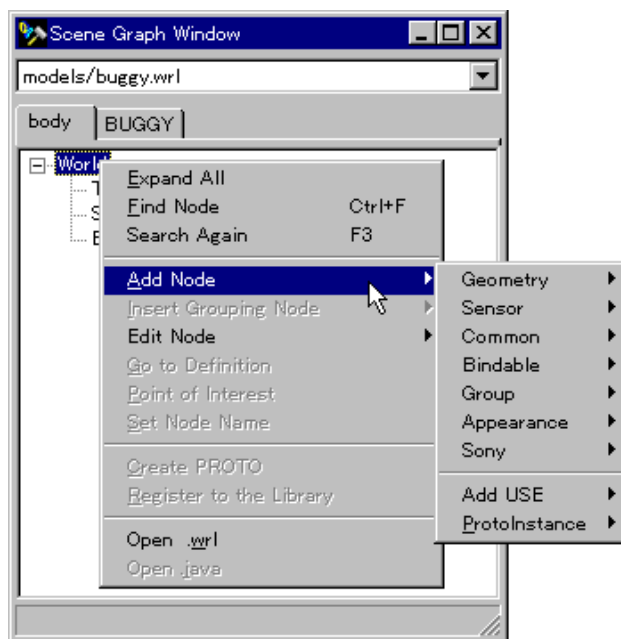


図 44 Add Node を選択している様子

Add USE

現在のスコープ内で有効な DEF 名が表示され、選択すると USE を使ってノードを生成します。

Add ProtoInstance

現在のスコープ内で有効な PROTO がサブメニューに表示され、選択するとインスタンスを追加します。

● Insert Grouping Node

選択されているノードに対する親ノードとして、グルーピングノードを追加します。

● Edit Node

選択されているノードのカット、コピー、ペースト、削除を行います。

- **Go to Definition**

PROTO インスタンスあるいは、Inline ノードが選択されている場合、定義スコープに切り替えます。

- **Point of Interest**

選択されているオブジェクトが正面に表示されるようにします。グルーピングノード、ジオメトリノード、ライトノード、Viewpoint に対してのみ有効です。

- **Set Node Name**

選択されているノードの DEF 名を設定するためのダイアログボックスを表示します。

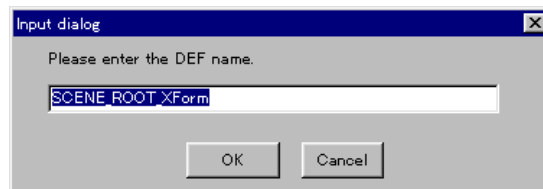


図 45 DEF 名設定用ダイアログ

- **Create PROTO**

選択されているノードをもとに PROTO を作成します。選択されているノードは PROTO インスタンスに置き換わります。

(注 1) このメニューを選択すると Undo できなくなります。

(注 2) CreatePROTO により PROTO を生成すると、自動的に大文字表記による DEF 名も付けられます。

- **Register to the Library**

選択されているノードが参照するファイルをライブラリーに登録します。登録が可能なファイルの候補がサブメニューに表示され、選択するとファイルのフォーマットに対応するライブラリーに登録されます。PROTO のスコープが選択されている場合は、PROTO 名がサブメニューに表示され、選択すると編集中の PROTO が Proto ライブリに登録されます。

同名のファイルが既に登録されている場合は、上書きを確認するダイアログボックスが表示されます。上書きしない場合は、続いて表示されるダイアログボックスで登録先のフォルダーを変更して下さい。

(注) ファイルのフォーマットについては、Library ウィンドウの章を参照して下さい。

- **Open .wrl**

選択されているファイルスコープの VRML ファイルを Text Editor に表示します。

(注) 編集中の VRML ファイルの内容を表示させるには、先に保存する必要があります。

- **Open .java**

選択されている Script ノードに対応する Java ファイルを Text Editor に表示します。Script ノードが選択されており、かつソースファイルが存在する場合にのみ、このメニューを使うことができます。

第 12 章 Script Expert ウィンドウ

VRML97 のスクリプトノードに対応した Java プログラムのテンプレート（ひな型）を生成します。

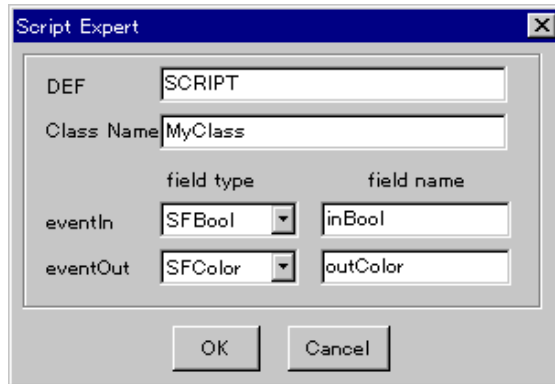


図 46 Script Expert ダイアログボックス

■ Script の生成法

以下の 4 つのフィールドに入力し、最後に OK ボタンを押します。

DEF:

VRML から Java プログラムを呼び出せるようにするために、スクリプトノードに名前をつけます。

Class Name:

Java のクラス名を指定します。

Java の仕様によりクラス名は、そのままファイル名になります。

field type:

eventIn または eventOut に対するフィールドの種類を指定します。VRML のフィールドの種類がすべてリストされますので、その中から選択してください。

field name:

eventIn または eventOut に対するフィールドの名前を指定します。

これにより、Java プログラムのテンプレート（ひな型）が作成され、Text Editor で表示されます。

また、同時に Scene Graph ウィンドウに Script ノードも追加されます。

（注）Script Expert ウィンドウでは複数の eventIn、eventOut を指定することはできません。複数の eventIn、eventOut をもつには、一度 Script Expert で Script を生成しておいて、後から、Attribute ウィンドウで追加してください。event の追加を行うには、Scene Graph ウィンドウで Script ノードを選択しておき、Attribute ウィンドウのポップアップメニューから、add field を選択します。

第13章 Text Editor ウィンドウ

図 47 に示す Text Editor ウィンドウは、VRML ファイルや Java ファイルを編集するためのテキストエディターです。スクリプトを編集している場合には、Attribute ウィンドウと連動して自動的に必要なハンドラが追加、削除されます。

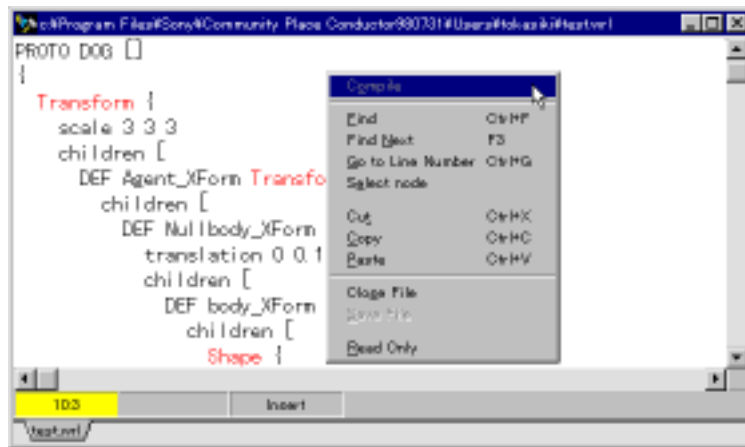


図 47 Text Editor ウィンドウ

■ Text Editor によるファイルの開き方

Text Editor は Scene Graph ウィンドウのポップアップメニュー（Open .wrl, Open .java）を選択するか、または Text Editor ウィンドウ内にファイルをドラッグ＆ドロップすることでファイルを開くことができます。

■ Text Editor の終了方法

Text Editor ウィンドウ右上の×ボタンをクリックすることで、終了させることができます。ファイルが変更されている場合には、保存を促すダイアログが表示されます。

(注) Scene Graph ウィンドウのポップアップメニューから Open .wrl を選択し、VRML ファイルを編集した場合は、ファイルを再読み込みするまで反映されません。

■ ポップアップメニュー

● Compile

編集中のファイルをコンパイルします。コンパイルが正常に終了したときは、Message ウィンドウに「Done」と表示されます。エラーが発生したときは、エラーメッセージが表示されます。

● Find

編集中のファイルに含まれる文字列を検索するためのダイアログボックスが表示されます。Replace、Replace All ボタンを押すことで、置換、全て置換を行うことも可能です。

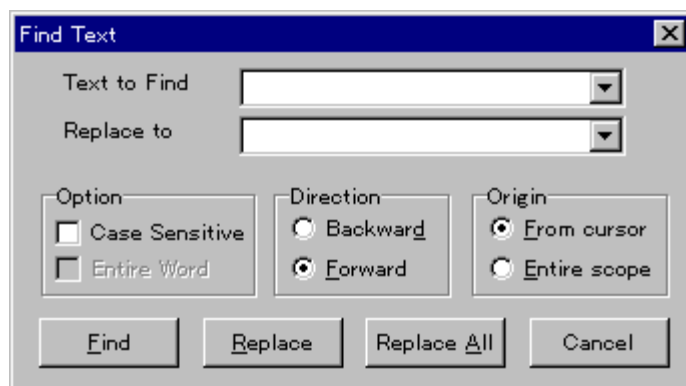


図 48 Find Text ダイアログボックス

Text to Find: 検索する文字列を入力します。

Replace to: 置換する文字列を入力します。

Option

Case Sensitive: チェックされている場合は大文字、小文字を区別します。

Entire Word: 完全に一致する文字列を検索します。

Direction

Backward: 逆方向へ検索します。
Forward: 順方向へ検索します。

Origin

From cursor: カーソルのある位置から検索します。
Entire scope: ファイルの頭から検索します。

• Find Next

Find で設定した条件の下で再検索します。

• Go to Line Number

指定した行へ移動します。

• Select node

VRML ファイル中の DEF 名を選択するとこのメニューが利用可能になります。Scene Graph ウィンドウのカレントスコープが一致する場合に、DEF 名がつけられたノードを選択状態にすることができます。

• Cut、Copy、Paste

テキストエリアのカット、コピー、ペーストを行います。

• Close File

編集集中のファイルを閉じます。

• Save File

編集集中のファイルを保存します。

• Read Only

編集集中のファイルの Read Only 属性を変更します。

第 14 章 Texture Mapping ウィンドウ

Texture Mapping ウィンドウでは、IndexedFaceSet ノードに対するテクスチャマッピングの指定および色の設定を行います。

(注) このウィンドウは IndexedFaceSet ノードを選択している場合にのみ使うことができます。

以下で各ページについて説明します。

■ Color ページ

Color タブをクリックすることにより図 49 に示す Color ページが表示されます。このページは、デフォルトでない Color Mode が選択された時にのみ表示されます。

Color ページには以下の情報が表示されます。



図 49 Color ページ



Face

Color Mode が「Use Appearance」以外に設定されている時、テクスチャが無い状態の現在の IndexedFaceSet が表示されます。

Color toolbar

表 15 に示すこれらのツールは Face 内でクリック、またはドラッグして使用します。

表 15 Color toolbar

ボタン	機能
 ブラシツール	面または頂点を現在の色でペイントします。
 スポイトツール	マウス位置に最も近い頂点の色を、読み取って現在の色として設定します。

Fill all faces

選択されている IndexedFaceSet のすべての面を、現在の色で塗りつぶします。

Current color

現在の色が表示されます。クリックすると色選択のダイアログボックスが表示されます。スポイトツールを使用することもできます。

Recently used colors

最近使った色が表示されます。クリックすることで再び使うことができます。

Change Color Mode

Change Color Mode ダイアログボックスを開きます。

Status bar

Color mode やツールによって異なったメッセージが表示されます。ブラシツールを使っているとき、以下の表 16 に示すメッセージが表示されます。

表 16 Color mode のメッセージ

カラーモード	メッセージ	意味
1 color per face (indexed)	Click on the face to use this color.	この色を使用するには面の上でクリックしてください。
1 color per face	Click on the face to use this color.	この色を使用するには面の上でクリックしてください。
1 color per vertex	Click near a vertex to change its color.	頂点の近くでクリックすると、その頂点の色が変わります。
1 color per vertex (indexed)	Click near a vertex to change its color.	頂点の近くでクリックすると、その頂点の色が変わります。

(注) 1 color per vertex、1 color per vertex (indexed) は頂点毎に色を設定するため、隣接する面にも影響があります。3D View ウィンドウでは正しい色が表示されません。

■ Texture Coordinate ページ

Texture Coordinates タブをクリックすることにより図 50 に示す Texture Coordinates ページが表示されます。このページには、デフォルトでない Texture Mode が選択された時にのみ表示されます。

Texture Coordinate ページには以下の情報が表示されます。

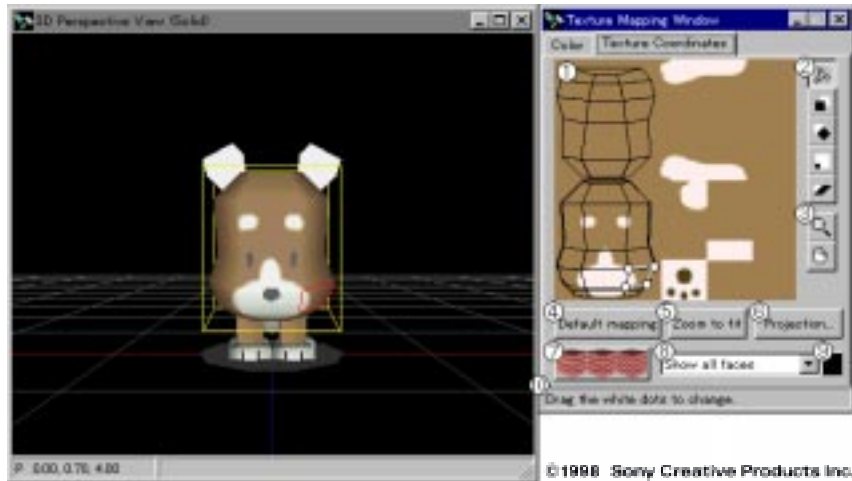


図 50 Texture Coordinate ページと 3D View ウィンドウ






Texture sample

3D/2D View ウィンドウで選択された IndexedFaceSet のテクスチャとテクスチャ座標が表示されます。選択されている IndexedFaceSet は、3D/2D View ウィンドウで赤枠で囲まれて表示されます。テクスチャが無いが、ロードできなかった場合、座標をセットするためにデフォルトのテクスチャが表示されます。

Texture coordinate tools

表 17 に示すテクスチャ座標を操作するツールがあります。これらのツールは Texture sample 内でマウスをドラッグして操作します。



表 17 Texture coordinate tools

ボタン	機能
 Point-by-point	テクスチャ座標の頂点を個別に移動します。
 Transform	テクスチャ座標を移動します。
 Rotation	白丸を中心にテクスチャ座標を回転します。白丸はドラッグで移動させることができます。
 Scale	テクスチャ座標を x、y 方向に拡大したり縮小します。
 Skew	テクスチャ座標を x、y 方向にゆがめます。

Texture sample tools

表 18 に示すこれらのツールはテクスチャ座標には何の効果も与えません。Texture sample で、テクスチャ座標の表示を操作する際に使います。

表 18 Texture sample tools

ボタン	機能
 Zoom	クリックした点が中心になるように、マウスの左ボタンでズームイン、右ボタンでズームアウトします。
 Pan	マウスをドラッグすることにより texture sample を移動します。

Default mapping

テクスチャ座標を囲む矩形が $[0, 0] - [1, 1]$ 内に合うようにテクスチャ座標を設定します。

Zoom to fit

テクスチャ座標全体が Texture sample 内におさまるように、Texture sample tools のズームとパンの設定を調整します。

Projection

Projection ダイアログボックスを開きます。

Change Texture Mode

Change Texture Mode ダイアログボックスを開きます。

Mesh Type

表 19 に示すように Mesh の表示範囲を指定します。

表 19 Mesh Type

種類	表示
Show selected face	選択した面のみ表示します。
Show neighbor faces	選択した面と隣接する面を表示します。
Show all faces	すべての面を表示します。

Mesh Color

Mesh Type が Show neighbor faces もしくは、Show all faces である場合、選択した Mesh 以外の Mesh の色を指定します。

Status bar

現在のツールに関するメッセージを表示します。

■ Change Color Mode ダイアログボックス

図 51 に示す Change Color Mode ダイアログでは IndexedFaceSet の *color*、*colorIndex*、*colorPerVertex* フィールドの設定を行います。デフォルトのモードは「Use Appearance」です。カラーモードはラジオボタンで選択できます。

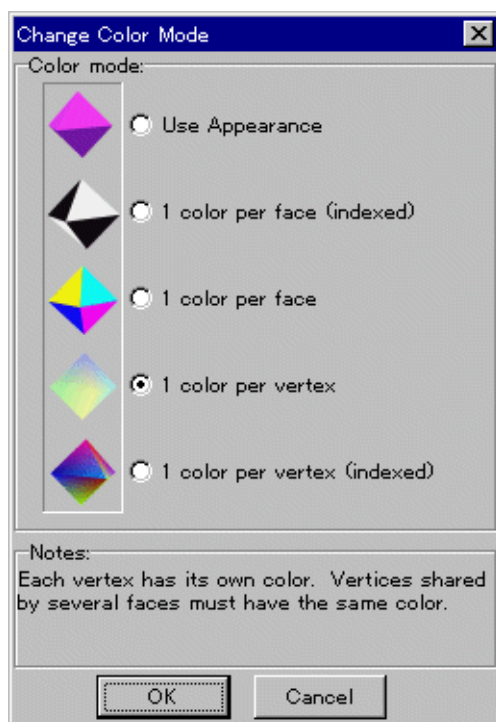


図 51 Change color mode ダイアログボックス

表 20 に示すカラーモード対応表は、各カラーモードと IndexedFaceSet のフィールドの関係を示しています。

表 20 カラーモード

カラーモード	<i>colorPerVertex</i> フィールド	<i>color</i> フィールド	<i>colorIndex</i> フィールド
Use Appearance	--	不使用	不使用
1 color per face (indexed)	FALSE	使用	使用
1 color per face	FALSE	使用	不使用
1 color per vertex	TRUE	使用	不使用
1 color per vertex (indexed)	TRUE	使用	使用

「Notes」の欄には、現在選択されているカラーモードの簡単な説明が表示されます。各カラーモードについての説明は表 21 のとおりです。

表 21 カラーモードの説明

カラーモード	表示内容	意味
Use Appearance	This IndexedFaceSet uses the color information given in its Appearance node.	Appearance ノードで与えられた色情報を使用します。
1 color per face (indexed)	Each face has a solid color, which is chosen from a palette of colors. This will optimize the file size if the number of different face colors is small (but greater than one) .	各面はカラーパレットから選ばれた単色で描画されます。異なる面の色数が少ない (ただし 1 以上) 場合は、1 color per face モードに比べ、こちらの方がファイルサイズは小さくなります。
1 color per face	Each face has a solid color. This will optimize the file size if the number of different face colors is large.	各面は単色で描画されます。異なる面の色数が多い場合は、1 color per face (indexed) モードに比べ、こちらの方がファイルサイズは小さくなります。
1 color per vertex	Each vertex has its own color. Vertices shared by several faces must have the same color.	各頂点ごとに色を割り当てます。複数の面によって共有されている頂点は同じ色を持ちます。
1 color per vertex (indexed)	Each vertex has its own color. Vertices shared by several faces may have different colors.	各頂点ごとに色を割り当てます。複数の面によって共有されている頂点の色が異なる場合があります。

(注) Change Color Mode にて 1 color per vertex あるいは、1 color per vertex (indexed) が選択されている場合は、3D View ウィンドウでは正しい色が表示されません。

以下に示すカラーモードの変更を行った場合、現在の色の情報は失われます。

1 color per face (indexed)	Use Appearance, 1 color per vertex
1 color per face	Use Appearance, 1 color per vertex
1 color per vertex	Use Appearance, 1 color per face(indexed), 1 color per face
1 color per vertex (indexed)	すべてのカラーモード

(注) OK ボタンをクリックしなければシーンは変化しません。

■ Change Texture Mode ダイアログボックス

図 52 に示す Change Texture Mode ダイアログボックスは IndexedFaceSet の *texCoord*、*texCoordIndex* フィールドの設定を行います。デフォルトのモードは「Default mapping」です。テクスチャモードはラジオボタンで選択できます。

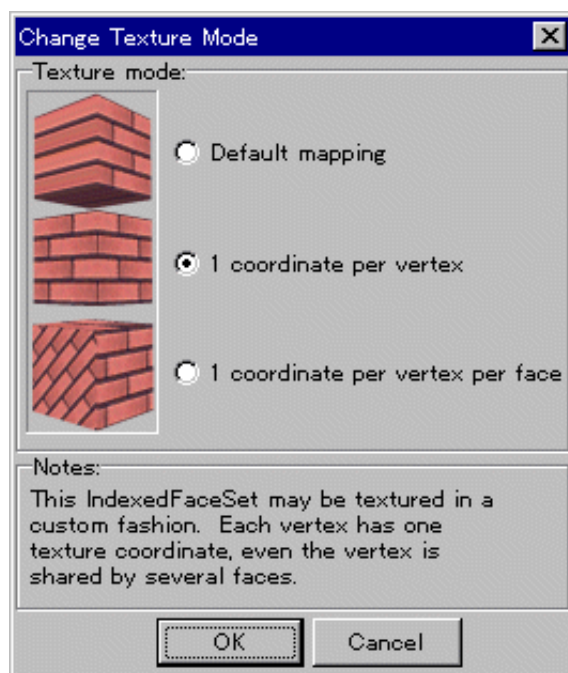


図 52 Change texture mode ダイアログボックス

表 22 に各テクスチャモードと IndexedFaceSet のフィールドとの関係について示しています。

表 22 テクスチャモードとフィールドの関係

テクスチャモード	<i>coord</i> フィールド	<i>coordIndex</i> フィールド
Default mapping	不使用	不使用
1 coordinate per vertex	使用	不使用
1 coordinate per vertex per face	使用	使用

「Notes」の欄には、現在選択されているテクスチャモードの簡単な説明が出ます。各テクスチャモードについての説明は以下のとおりです。

表 23 テクスチャモードの説明

テクスチャモード	表示内容	意味
Default mapping	This IndexedFaceSet will be textured automatically using the method given in the VRML specification.	VRML の仕様で与えられる方法を用いて自動的にテクスチャが貼られます。
1 coordinate per vertex	This IndexedFaceSet may be textured in a custom fashion. Each vertex has one texture coordinate, even the vertex is shared by several faces.	各頂点は、複数の面に共有されている場合でも、一つのテクスチャ座標を持ちます。
1 coordinate per vertex per face	This IndexedFaceSet may be textured in a custom fashion. Each vertex has one texture coordinate for each face that uses it.	各頂点は、それを使用する各面ごとに一つのテクスチャ座標を持ちます。

カラーモードの場合と同様に、テクスチャモードが変わるとオブジェクトの見た目が変わる場合があります。オブジェクトの見た目が変化するのは以下に示す場合です。

1 color per vertex	Default mapping
1 coordinate per vertex per face	すべてのカラーモード

(注) OK ボタンをクリックしなければシーンは変化しません。

■ Texture Mapping Projection ダイアログボックス

図 53 に示す Texture Mapping Projection ダイアログボックスでは、各投影法に基づいてすべてのテクスチャ座標を自動的に設定できます。

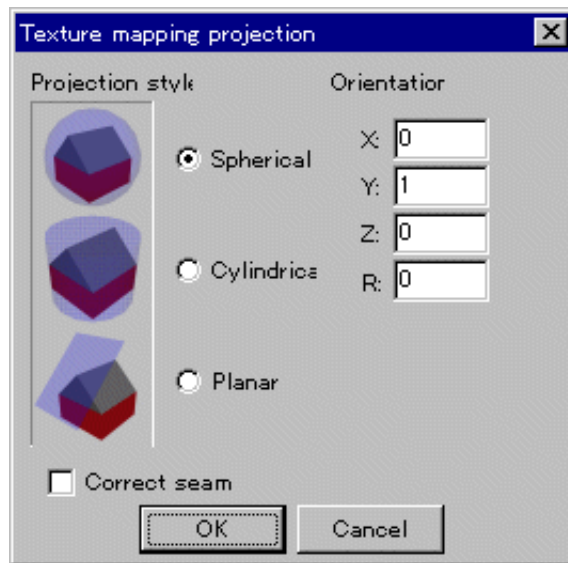


図 53 Texture Mapping Projection ダイアログボックス

投影法には表 24 に示す 3 つの種類があります。

表 24 投影法の説明

投影法	説明
Spherical	VRML の仕様にしたがってテクスチャマッピングされた IndexedFaceSet を囲む仮想的な球 (Shphere ノード) を s とします。IndexedFaceSet の各頂点 v について、 v と s の中心を通る線を l とします。 v のテクスチャ座標は、 s と l の交点のテクスチャ座標と同じになります。
Cylindrical	VRML の仕様にしたがってテクスチャマッピングされた IndexedFaceSet を囲む仮想的な円柱 (Cylinder ノード) を c とします。IndexedFaceSet の各頂点 v について、 v と c の中心軸を通り中心軸に直交する線を l とします。 v のテクスチャ座標は、 c と l の交点のテクスチャ座標と同じになります。
Planar	VRML の仕様にしたがってテクスチャマッピングされた仮想的な平面 (IndexedFaceSet ノード) を p とします。IndexedFaceSet の各頂点 v について、 v のテクスチャ座標は、 v に最も近い p 上の点のテクスチャ座標と同じになります。

投影に使用する仮想的な形状の向きは、投影する形状の VRML スタイルの回転 (x, y, z, r) で与えます。OK ボタンを押すと、現在の形状に投影されます。

「Correct seam」は spherical または cylindrical 投影に有用です。この投影法の場合、面がテクスチャのつなぎ目にくることがあるので、形状の左側の s 座標が右側より大きい場合があります。この問題を形状の右側に 1.0 を加えて解決します。

(注) 1 coordinate per vertex モードの場合は、一つの頂点が二つの異なる値を持つ必要があるため、正しく機能しません。

第 15 章 Object Customize Wizard の作成方法

Conductor には通常の Attribute ウィンドウよりも簡単に、オブジェクトの属性を編集できる Object Customize Wizard という機能が用意されています。選択しているオブジェクトが Wizard に対応している場合、Attribute ウィンドウに Wizard タブが表示され、スライダーバー、ラジオボタン、コンボボックス、カラーピッカーで構成される UI によって直感的に属性を編集することができます。



図 54 Attribute ウィンドウ

■ VRML オブジェクトの変更

VRML オブジェクトを Wizard 対応にするには、以下の手順にしたがって、VRML オブジェクトに変更を加え、Wizard の設定ファイルを用意してください。処理を行うウィンドウまたは、ポップアップメニューの選択順はカッコ内に示します。

1. Script ノードを作る。(Scene Graph Window Add Node Common Script)
2. Script ノードに MFString 型の wizInfo というフィールドを追加する。
(Attribute Window add field)
3. wizInfo フィールドに、後で作成する Wizard 設定ファイルの URL を指定する。(Attribute Window)
4. 1. で作成した Transform ノードを PROTO で定義する。(Scene Graph Window Create PROTO)
5. Scene Graph Window のコンボボックスで、カレントスコープを PROTO 定義部 (PROTO のスコープ) にする。Wizard で変更するフィールドと、wizInfo フィールドを IS マッピングする。(Attribute Window map with IS)
6. PROTO インスタンスのあるタブ (body タブ) に戻り、すべてのノードを消去し保存する。

■ Wizard の設定ファイルの記述

Wizard の設定ファイルの記述は Text Editor を用いて以下のように行います。

Wizard の Window には UI が上から順に並びます。設定ファイルには、その順に上から UI の設定を記述します。例えば、スライダーバーを Wizard の一番上に表示したいときは、設定ファイルの一番上に [slider] と記述し、それに続けてスライダーバーの最大値などを記述します (「スライダーバーの設定」(87 ページ) 参照)。ラジオボタンを表示したいときは、[radio] と記述し、ラジオボタンに必要な設定を記述します (「ラジオボタンの設定」(88 ページ) 参照)。

ただし 1 つの Wizard に同じ種類の UI が複数ある場合、例えばスライダーバーが 2 つある場合、「slider」の後ろに文字列を付け加え、[slider.1]、[slider_height] などとして設定ファイル内で名前が重ならないようにしてください。[] 内の文字列が「slider」で始まれば、スライダーとみなされます。他の UI に関しても同様です。

現在使用できる UI と識別子を表 25 に示します。

表 25 UI と識別子の対応関係

UI の種類	識別子
スライダーバー	[slider]
ラジオボタン	[radio]
コンボボックス	[combo]
カラーピッカー	[color]

• スライダーバーの設定

例 -----

```
[slider.1]
name=height
caption="Seat Height"
value=32
min=10
max=80
```



表 26 スライダーバーにおける変数の説明

変数	説明
name	設定ファイル内での、スライダーバーの名前です。 設定ファイル内で同じ名前が存在してはいけません。 空白文字を含んではいけません。
caption	Wizard に表示される UI の名前です。 日本語も使用可能です。 空白文字を含む場合は文字列をダブルクォート(")で囲みます。
value	スライダーバーの初期値です。整数のみ使用可能です。
min	スライダーバーの最小値です。整数のみ使用可能です。
max	スライダーバーの最大値です。整数のみ使用可能です。

• ラジオボタンの設定

例 -----

```
[radio_2]
name=shape
caption="Type of Seat"
value=0
option="rectangle","circle"
```



表 27 ラジオボタンにおける変数の説明

変数	説明
name	設定ファイル内での、ラジオボタンの名前です。 設定ファイル内で同じ名前が存在してはいけません。 空白文字を含んではいけません。
caption	Wizard に表示される UI の名前です。 日本語も使用可能です。 空白文字を含む場合は文字列をダブルクォート(")で囲みます。
value	ラジオボタンの初期値です。 選択肢(option)の1番が0、2番が1、というように対応します。
option	ラジオボタンで選べる各選択肢の説明文です。 各文字列はカンマ(,)で区切ります。 日本語も使用可能です。 空白文字を含む場合は文字列をダブルクォート(")で囲みます。

• コンボボックスの設定

例 -----

```
[combo1]
name=shape
caption="Shape of Seat"
value=0
```



```
option="rectangle", "circle"
```



ラジオボタンの設定と同じです。

• カラーピッカーの設定

例 -----

```
[color]
name=color
caption="Color of Seat"
value=0.5 0.8 0.5
```

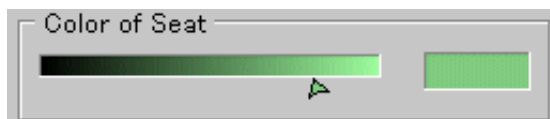


表 28 カラーピッカーにおける変数の説明

変数	説明
name	この設定ファイル内での、コンボボックスの名前です。この設定ファイル内で同じ名前が存在してはいけません。空白文字を含んではいけません。
caption	Wizard に表示される UI の名前です。 日本語も使用可能です。 空白文字を含む場合は文字列をダブルクォート(")で囲みます。
value	カラーピッカーの初期値です。 VRML と同じ RGB 形式で記述します。

• Wizard の各 UI と VRML フィールドの対応付け

Wizard の UI で入力した値を VRML に反映させるためには、UI と VRML フィールドの対応を記述する必要があります。

UI とフィールドの対応付けは、[route] と記述した後に、基本的には左辺を VRML フィールド名、右辺を UI に付けた名前 (name) とした代入式として記述します。この記述は、上述の各 UI の設定の記述より後 (ファイルの最後) に行ってください。

例 -----

```
[route]
seatTrans.y= (height - 32 ) / 100
poleScale.y=height * 2 / 64
seatType=shape
seatColor=color
-----
```

左辺には PROTO のインタフェース部分に定義されているフィールドのうち、SFFloat、SFInt32、SFColor、SFVec3f、SFRotation 型のフィールドを記述できます。

ただし、SFVec3f、SFRotation 型は各成分に分解する必要があります。

SFVec3f 型は xyz 成分に分解して記述します。例えば、上の例のように「seatTrans.y」と記述することで、seatTrans フィールド (SFVec3f 型) の y 成分に右辺の値をセットできます。

SFRotation 型は x、y、z、angle に分解して、例えば、「seatRot.angle」というように記述します。ただし SFRotation 型は「angle」をデフォルトとしてあるので、この場合は「.angle」を省略して「seatRot」のみでも構いません。

右辺には左辺の VRML フィールドに値をセットするために用意した UI の名前を記述します。

左辺が SFFloat、SFInt32、SFVec3f、SFRotation 型の場合、右辺には数式を記述することが出来ます。数式には UI の名前、定数、四則演算子、カッコが使用できます。

なお、UI と VRML フィールドは 1 対 1 に対応している必要はありません。つまり、1 つの UI から複数の VRML フィールドへ値をセットすることができます。例では UI の値 height から seatTrans.y と poleScale.y に値をセットしています。

- **その他**

Conductor を起動した時に、Library ウィンドウの PROTO タブにあらわれる VRML ファイルは、デフォルトでは Conductor をインストールしたフォルダーの datalib/proto/ 直下にある VRML ファイルのみです。datalib/proto/にあるフォルダー以降の VRML ファイルは Library ウィンドウには表示されません。

VRML ファイルと同じ名前で、拡張子が .bmp のビットマップファイルが同じフォルダーにある場合、Library ウィンドウでアイコンとして使用されます。無い場合はデフォルトアイコンが使用されます。

スクリプトリファレンス

ライブラリに含まれるスクリプトの詳細を説明します。

表 29 ライブラリに含まれるスクリプト

Basic Scripts		Movement Scripts	Other Scripts
<u>CPSCNot</u>	<u>CPSCBoolVec</u>	<u>CPSCMoveOn</u>	<u>CPSCShowBool</u>
<u>CPSCAnd</u>	<u>CPSCBoolRot</u>	<u>CPSCMoveIt</u>	<u>CPSCShowFloat</u>
<u>CPSCOr</u>	<u>CPSCEdge</u>	<u>CPSCMoveAlong</u>	<u>CPSCShowInt</u>
<u>CPSCXor</u>	<u>CPSCStep</u>	<u>CPSCMoveTo</u>	<u>CPSCShowVec</u>
<u>CPSCSineWave</u>	<u>CPSCUpper</u>	<u>CPSCRotateAlong</u>	<u>CPSCReturnBool</u>
<u>CPSCToggle</u>	<u>CPSCLower</u>	<u>CPSCRotateTo</u>	
<u>CPSCToggleFrac</u>	<u>CPSCMatch</u>	<u>CPSCFace</u>	
<u>CPSCBoolToTime</u>	<u>CPSCRange</u>		
<u>CPSCFloatToInt</u>	<u>CPSCDelayGen</u>		

CPSCNot

```

DEF CPSCNot Script {
    url "scripts/CPSCNot.class"
    eventIn SFBool inBool
    eventOut SFBool outBool
    field SFString info "this is CPSCNot script."
}

```

inBool に対して NOT 論理演算を行い、その結果を出力します。TRUE が与えられれば FALSE を、FALSE が与えられれば TRUE を出力します。

CPSCAnd

```
DEF CPSCAnd Script {  
    url "scripts/CPSCAnd.class"  
    eventIn SFBool inOne  
    eventIn SFBool inTwo  
    eventOut SFBool outBool  
    field SFString info "this is CPSCAnd script"  
}
```

このスクリプトは (inOne、inTwo) が与えられられると、それらの AND 論理演算を行い、結果を出力します。両方が TRUE だった場合には TRUE を、それ以外の場合には FALSE を出力します。

CPSCOr

```
DEF CPSCOr Script {  
    url "scripts/CPSCOr.class"  
    eventIn SFBool inOne  
    eventIn SFBool inTwo  
    eventOut SFBool outBool  
    field SFString info "this is CPSCOr script"  
}
```

(inOne、inTwo) が与えられると、それらの OR 論理演算を行い、結果を出力します。両方が FALSE だった場合には FALSE を、それ以外の場合には TRUE を出力します。

CPSCXor

```
DEF CPSCXor Script {  
    url "scripts/CPSCXor.class"  
    eventIn  SFBool inOne  
    eventIn  SFBool inTwo  
    eventOut SFBool outBool  
    field SFString info "this is CPSCXor script"  
}
```

(inOne、inTwo) が与えられると、それらの XOR (排他的論理和) 論理演算を行い、結果を出力します。inOne と inTwo が一致していた場合には FALSE を、一致しなかった場合には TRUE を出力します。

CPSCSineWave

```
DEF CPSCSineWave Script {  
    url "scripts/CPSCSineWave.class"  
    eventIn  SFFloat inFloat  
    eventIn  SFFloat phase  
    eventIn  SFFloat amplitude  
    eventIn  SFFloat waveLength  
    eventOut SFFloat outFloat  
    field SFString info "this is CPSCSineWave script"  
}
```

与えられた浮動小数の正弦値を出力します。ここで、phase は位相 (0.0) を、amplitude は振幅 (1.0) を、waveLength は波長 (1.0) を表します。

(注) カッコ内の数値は、デフォルト値です。

CPSCToggle

```
Def CPSCToggle Script {  
    url "scripts/CPSToggle.class"  
    eventIn    SFBool  inBool  
    eventOut   SFBool  outBool  
    field      SFBool  defBool TRUE  
    field SFString    info "this is CPSCToggle script."  
}
```

トグルスイッチです。スクリプトに TRUE が与えられるたびに、TRUE と FALSE を交互に出力します。スクリプトが最初に出力する値は `defBool` で定義でき、デフォルトでは TRUE になっています。

CPSCToggleFrac

```
Def CPSCToggleFrac Script {  
    url "scripts/CPSToggleFrac.class"  
    eventIn    SFFloat  inFloat  
    eventOut   SFFloat  outFloat  
    field SFString    info "this is CPSCToggleFrac  
script."  
}
```

このスクリプトはインターポレーターをトグルするのに使います。初期状態ではスクリプト内の変数 `forward` は TRUE になっています。入力された `inFloat` が減少すると、`forward` が TRUE と FALSE をトグルします。ここで、もし `forward` が TRUE であれば、`outFloat` は `inFloat` と同じ値を出力します。FALSE であれば、`outFloat` は `1-inFloat` を出力します。

CPSCBoolToTime

```
DEF CPSCBoolToTime Script {  
    url "scripts/CPSCBoolToTime.class"  
    eventIn  SFBool inBool  
    eventOut SFTIME outTime  
    field SFString info "this is CPSCBoolToTime script."  
}
```

スクリプトが TRUE イベントを受け取ると、そのイベントのタイムスタンプを出力します。同時に、Java Console にタイムスタンプの値を double 型で出力します。

CPSCFloatToInt

```
DEF CPSCFloatToInt Script {  
    url "scripts/CPSCFloatToInt.class"  
    eventIn  SFFloat inFloat  
    eventOut SFInt32 outInt  
    field SFString info "this is CPSCFloatToInt script."  
}
```

このスクリプトは与えられた浮動小数を四捨五入した整数値を出力します。

CPSCBoolVec

```
DEF CPSCBoolVec Script {  
    url "scripts/CPSCBoolVec.class"  
    eventIn  SFBool inBool  
    eventOut SFVec3f outVec  
    field SFVec3f vector 0 0 0  
    field SFString info "this is CPSCBoolVec script."  
}
```

スクリプトが TRUE イベントを受け取ると、vector フィールドで定義された SFVec3f 値を出力します。デフォルトでは、(0, 0, 0) を出力するようになっています。

CPSCBoolRot

```
DEF CPSCBoolRot Script {  
    url "scripts/CPSCBoolRot.class"  
    eventIn  SFBool    inBool  
    eventOut SFRotation outRot  
    field SFRotation    rotate 0 0 1 0  
    field SFString      info  "this is CPSCBoolRot  
script."  
}
```

スクリプトが TRUE を受け取ると、rotate フィールドで定義された SFRotation 値を出力します。デフォルトでは、(0 0 1 0) を出力するようになっています。

CPSCEdge

```
DEF CPSCEdge Script {  
    url "scripts/CPSCEdge.class"  
    eventIn  SFBool    inBool  
    eventOut SFBool    outBool  
    field    SFString info  "this is CPSCEdge script"  
}
```

このスクリプトは入力の値が変わるたびに TRUE または FALSE を交互に出力します。入力に変化しなかった場合（例えば、TRUE が連続して与えられた場合）スクリプトは値を出力しません。

CPSCStep

```
DEF CPSCStep Script {
    url "scripts/CPSCStep.class"
    eventIn  SFFloat  inBool
    eventOut SFFloat  outFloat
    field    SFFloat  lower 0.0
    field    SFFloat  upper 10.0
    field    SFFloat  step 1.0
    field    SFString info  "this is CPSCStep script"
}
```

スクリプトに TRUE が入力されるたびに、`lower` で定義された値から `upper` で定義された値まで、`step` ずつ増加させた浮動小数を出力します。`upper` を超えると、それ以上増加はしなくなります。

例えば、デフォルト値をそのまま使用すると、最初に TRUE を受け取ったときスクリプトは 0.0 を出力します。次に TRUE を受け取ると 1.0 を出力し、そしてその次は 2.0.....と続きます。しばらくして 10.0 を出力すると、それ以上は増加しなくなり、TRUE を受け取るたびに 10.0 を出力し続けます。

また、`upper`、`lower`、`step` に定義されている値と現在出力されている値は Java Console に表示されます。

CPSCUpper

```
DEF CPSCUpper Script {
    url "scripts/CPSCUpper.class"
    eventIn  SFFloat  inFloat
    eventOut SFFloat  outBool
    field    SFFloat  limit 5.0
    field    SFString info  "this is CPSCUpper script"
}
```

`limit` フィールドで定義されている値よりも大きな値が入力されると TRUE を出力します。入力された値が `limit` に等しければ、値は出力しません。

`limit` フィールドのデフォルト値は 5.0 となっており、この場合 5.0 より大きい値（例えば 6.0）が入力されると TRUE が出力されます。

CPSCLower

```
DEF CPSCLower Script {  
    url "scripts/CPSCLower.class"  
    eventIn  SFFloat  inFloat  
    eventOut SFBool   outBool  
    field    SFFloat  limit 5.0  
    field    SFString info  "this is CPSCLower script"  
}
```

このスクリプトは `limit` フィールドで定義されている値よりも小さな値が入力されると `TRUE` を出力します。入力された値が `limit` に等しければ、値を出力しません。

`limit` フィールドのデフォルト値は5.0となっており、この場合5.0より小さい値（例えば4.0）が入力されると `TRUE` が出力されます。

CPSCMatch

```
DEF CPSCMatch Script {  
    url "scripts/CPSCMatch.class"  
    eventIn  SFFloat  inFloat  
    eventOut SFBool   outBool  
    field    SFFloat  value 5.0  
    field    SFString info  "this is CPSCMatch script"  
}
```

このスクリプトは `value` フィールドで定義されている値と等しい値が入力されると `TRUE` を出力します。入力された値が `value` に等しくなければ、値を出力しません。

`value` フィールドのデフォルト値は5.0となっており、この場合5.0が入力されると `TRUE` が出力されます。

CPSCRange

```
DEF CPSCRange Script {  
    url "scripts/CPSCRange.class"  
    eventIn  SFFloat  inFloat  
    eventOut SFBool   outBool  
    field    SFFloat  upperLimit 6.0  
    field    SFFloat  lowerLimit 4.0  
    field    SFString info    "this is CPSCRange script"  
}
```

スクリプトに入力された値が `lowerLimit` 以上 `upperLimit` 以内の範囲にあるとき、TRUE を出力します。値がその範囲内に存在しない場合は、スクリプトは値を出力しません。

`lowerLimit` と `upperLimit` のデフォルト値は、4.0 と 6.0 となっています。この場合、4.0 以上 6.0 以下の値が入力された場合に、TRUE が出力されます。

CPSCDelayGen

```
DEF CPSCDelayGen Script {  
    url "scripts/CPSCDelayGen.class"  
    eventIn  SFTIME    inTime  
    eventOut SFTIME    outTime  
    field    SFFloat   delay 5  
    field    SFString  info    "this is CPSCDelayGen script"  
}
```

SFTIME イベントを受け取ると、このスクリプトは受け取った値に `delay` フィールドで指定された浮動小数を加えた値を出力します。`delay` フィールドのデフォルト値は 5.0 となっています。

CPSCMoveOn

```
DEF CPSCMoveOn Script {  
    url "scripts/CPSCMoveOn.class"  
    eventIn  SFFloat      inFloat  
    eventOut SFVec3f      posChange  
    eventOut SFRotation   oriChange  
    field    SFString     info    "this is CPSCMoveOn  
script"  
}
```

オブジェクトの動作と回転を定義します。
デフォルトでは正弦波を描くようになっていますが、java ソースコードを編集することによって動作を定義できます。

CPSCMoveIt

```
DEF CPSCMoveIt Script {  
    url "scripts/CPSCMoveIt.class"  
    directOutput TRUE  
    eventIn  SFBool      clicked  
    field    SFNode      node NULL  
    field    SFString     info    "this is CPSCMoveIt script"  
}
```

このスクリプトは TRUE が入力されるたびにオブジェクトを特定の場所(スクリプト内で定義します)に移動させます。

CPSCMoveAlong

```
DEF CPSCMoveAlong Script {
    url "scripts/CPSCMoveAlong.class"
    directOutput TRUE
    eventIn SFVec3f    position
    field   SFNode     node NULL
    field   SFString   info    "this is CPSCMoveAlong
script"
}
```

ノードを新しい位置に `position` 分だけ移動させます。
Position Interpolatorとあわせて使うことによって、インターポレーターに沿ってノードを動かすことができます。

CPSCMoveTo

```
DEF CPSCMoveTo Script {
    url "scripts/CPSCMoveTo.class"
    directOutput TRUE
    eventIn SFFloat    fraction
    eventIn SFVec3f     position
    field   SFNode     node NULL
    field   SFString   info    "this is CPSCMoveTo script"
}
```

スクリプトが `fraction` を受け取ると、`node` フィールドで指定されたオブジェクトを `position` フィールドで定義された場所に向かって移動させます。`position` に新しい値が指定されなければ、オブジェクトは移動しません。

(注) **translation** フィールドが複数存在する場合、スクリプトがどこを参照しているかによって予想しない方向へ移動してしまうことがあります。ノードをマウスでドラッグ移動すると新しい `translation` が作られます。

CPSCRotateAlong

```
DEF CPSCRotateAlong Script {  
    url "scripts/CPSCRotateAlong.class"  
    directOutput TRUE  
    eventIn  SFRotation    orientation  
    field    SFNode        node NULL  
    field    SFString      info    "this is CPSCRotateAlong  
script"  
}
```

このスクリプトはノードを、相対的に新しいorientationに回転させます。Orientation Interpolator と合わせて使うことによって、ノードを Interpolator に沿って動かすことができます。

CPSCRotateTo

```
DEF CPSCRotateTo Script {  
    url "scripts/CPSCRotateTo.class"  
    directOutput TRUE  
    eventIn  SFFloat        fraction  
    eventIn  SFRotation     orientation  
    field    SFNode        node NULL  
    field    SFString      info    "this is CPSCRotateTo  
script"  
}
```

スクリプトに新しい fraction が入力されると、node フィールドで指定されたオブジェクトを orientation で定義された方向に回転させます。新しい値が orientation に与えられなければ、オブジェクトは回転しません。

CPSCFace

```
DEF CPSCFace Script {  
    url "scripts/CPSCFace.class"  
    directOutput TRUE  
    eventIn SFVec3f      inVec  
    eventIn SFRotation   outRot  
    field SFNode         node NULL  
    field SFString       info  "this is CPSCFace script"  
}
```

スクリプトに新しい `inVec` が入力されると、`node` フィールドで指定されたオブジェクトを `inVec` の位置に向くように（例えば、円錐の場合はそちらを指すように）回転させます。

（注）オブジェクトを正しく回転させるために、回転させるオブジェクトの Transform ノードを確認して下さい。一度新しい Transform ノードを作成してそちらを参照させることもできます。

CPSCShowBool

```
DEF CPSCShowBool Script {  
    url "scripts/CPSCShowBool.class"  
    eventIn SFBool   inBool  
    eventOut SFBool  outBool  
    field SFString info  "this is CPSCShowBool script"  
}
```

受け取った論理値をそのまま出力し、Java Console にも表示します。このスクリプトは、スクリプトなどに正しい値が渡されているのかを調べるために使うことができます。

CPSCShowFloat

```
DEF CPSCShowFloat Script {  
    url "scripts/CPSCShowFloat.class"  
    eventIn  SFFloat  inFloat  
    eventOut SFFloat  outFloat  
    field    SFString info    "this is CPSCShowFloat  
script"  
}
```

受け取った浮動小数値をそのまま出力し、Java Console にも表示します。このスクリプトは、スクリプトなどに正しい値が渡されているのかを調べるために使うことができます。

CPSCShowInt

```
DEF CPSCShowInt Script {  
    url "scripts/CPSCShowInt.class"  
    eventIn  SFInt32  inInt  
    eventOut SFInt32  outInt  
    field    SFString info    "this is CPSCShowInt script"  
}
```

受け取った整数値をそのまま出力し、Java Console にも表示します。このスクリプトは、スクリプトなどに正しい値が渡されているのかを調べるために使うことができます。

CPSCShowVec

```
DEF CPSCShowVec Script {  
    url "scripts/CPSCShowVec.class"  
    eventIn  SFVec3f  inVec  
    eventOut SFVec3f  outVec  
    field    SFString info    "this is CPSCShowVec script"  
}
```

受け取った3次元ベクトル値をそのまま出力すると同時に Java Console にも出力します。このスクリプトは、スクリプトなどに正しい値が渡されているのかを調べるために使うことができます。

CPSCReturnBool

```
DEF CPSCReturnBool Script {
    url "scripts/CPSCReturnBool.class"
    eventIn SFBool    inBool
    eventOut SFBool    outBool
    field    SFBool    retBool TRUE
    field    SFString info    "this is CPSCReturnTrue
script"
}
```

スクリプトに論理値が与えられると、入力された値には関係なく、`retBool` フィールドで定義された値を出力します。