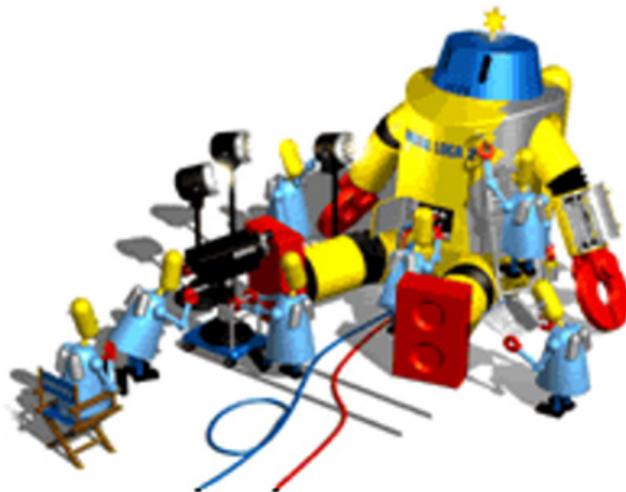




DigitalLoca3 for Stylos

チュートリアルガイド



■本チュートリアルについて

本ドキュメントは、「DigitalLoca3」(*)を対象として作成されたチュートリアルを、「DigitalLoca3 for Stylos」をお使いのユーザーのために提供させていただくものです。

本書をお読みいただくことによって、「DigitalLoca3 for Stylos」の基本的な操作方法をご理解いただくことができます。

なお、本ドキュメントは「DigitalLoca3」を対象として作成されておりますため、ご利用にあたっては以下の点にご注意くださいますようお願い申し上げます。

- 「DigitalLoca3 for Stylos」は、「DigitalLoca3」の持つ機能の一部がご利用いただけません。詳細については『「DigitalLoca3 for Stylos」の機能制限について』をご参照ください。
- 本文中に登場する「ユーザーインターフェース」「コマンド名」等の画面写真は、「DigitalLoca」の、本ドキュメント作成時点でのバージョンに基づいておりますため、内容が若干異なる場合があります。

※・・・「DigitalLoca3」は、株式会社ネットディメンションが開発・販売するソフトウェアです。Stylos にバンドルされる「DigitalLoca3 for Stylos」と「DigitalLoca3」は異なる商品ですので、ご注意ください。

■ 「DigitalLoca3 for Stylos」の機能制限について

「DigitalLoca3 for Stylos」は、「DigitalLoca3」の持つ機能の中からアニメーションの作画作業に必要な機能のみを選び出した、「Stylos」専用のソフトウェアです。

「DigitalLoca3」の機能うち、「DigitalLoca3 for Stylos」でご利用いただけない機能は、下記のとおりです。

	DigitalLoca3	DigitalLoca3 for Stylos
ランタイムファイル(*.LCR形式、*.EXE)の出力	○	×
キャビネットファイル(*.LCO)の出力	○	×
アイコンファイル(*.ICO)の出力	○	×
XAML形式のエクスポート	○	×
キャストを素材(DXF,BMP,WAV,TXT,AVI)として書き出す	○	×

DigitalLoca チュートリアル

目次

チュートリアルガイド

DigitalLoca へようこそ！	6
DigitalLoca とは？	7
DigitalLoca 基本用語	8
「プロジェクト」	8
「3次元ワールド」	8
「モデリング」	8
「テクスチャ」	8
「キャスト」	9
「アニメーション」	9
作業に入る前に	10
1章では	11
モデルキャストを作成する	12
モデルキャストの質感を設定する	13
色と質感を付ける	14
テクスチャを貼り付ける	16
モデルキャストの形状を編集する	21
モデリングビューウィンドウの編集画面	22
モデリングビューウィンドウで形状を編集する	23
プロジェクトを保存する	42
モデルをキャストごとに分割する	43
モデルの中心を変更する	49
各キャストに素材を設定する	51
モデルキャストを配置する	53
親子リンクを設定する	57
カメラキャストを配置する	61
ライトキャストを配置する	63
プロジェクトの再生、保存	65
2章では	67
モデルキャストにアニメーションを設定する	68
複数のキーフレームを作成する	68
アニメーションに効果音を付ける	74
複数のアニメーションを同時に再生する	77
マルチスコア機能を使用する	81
マルチスコア機能について	83

DigitalLoca チュートリアル

3章では	85
キャストクリックイベントを使用する	86
マウスクリックに反応させる	86
キャビネットファイルを使用する	95
別プロジェクトに移動する	95
当り判定と条件分岐命令を使用する	102
当り判定と条件分岐	102
DigitalLoca チュートリアルガイド終了!	109

スクリプト入門

第1章: デジタルロケにおけるスクリプト	112
・ はじめに	112
・ 学習を進めていく上で	112
1-1. スクリプトとは	112
1-2. イベントとは	113
1-3. スコアとの関係	114
1-4. キャスト名	115
1-5. 処理	115
・ 章末	115
第2章: スクリプトの基礎	117
・ はじめに	117
・ 章構成	117
2-1. 変数・定数	117
2-2. 型	118
2-3. 変数・定数を使うには	118
2-4. グローバルとローカル	119
2-5. 手続き	121
2-6. 引数	124
2-7. 関数と戻り値	124
・ 章末	126
第3章: 代表的な命令	127
・ はじめに	127
・ 本章で学ぶこと	127
3-1. キー情報の取得	127
3-2. 条件分岐命令	128
3-3. パペット操作	130
3-4. 衝突判定	132
・ 章末	133

第 4 章:コンテンツを作ろう	134
・はじめに	134
・ サンプルコンテンツ	134
・ ストーリー	134
・ コンテンツの流れ.....	134
4-A. 初期化処理.....	136
4-B1. 移動処理	140
4-B2. カギ処理	145
4-B3. ドア処理	146
4-B4. 視点切換処理	148
・ まとめ.....	149
・ 最後に.....	149



DigitalLoca3 for Stylos

チュートリアルガイド



DigitalLoca へようこそ！

DigitalLoca の世界へようこそ。

このチュートリアルガイドは、読みながら手順通りに作業を進めていくうちに、作品を制作する上での基本的な操作方法が学べるように考慮されています。このチュートリアルガイドで、DigitalLoca の全ての機能を学ぶことはできませんが、はじめに本マニュアルを読みながら作業を進めて頂ければ、使用方法を覚える為の一番の近道となるでしょう。

このチュートリアルガイドを読みながら実際に作業を行い、最終的に 1 つのサンプル作品を作りあげてください。

DigitalLoca で制作できる作品のほんの一例ではございますが、これから皆様が作られる作品の参考になればと思います。



チュートリアルガイドでの制作作品完成図

DigitalLoca とは？

近年、様々なマルチメディアオーサリングツールが登場し、それによってユーザーの方々が、比較的簡単にコンピュータの中で作品を制作することができるようになりました。本アプリケーションは、その数あるオーサリングツールの中でも、絵やテキスト、サウンド、プログラムといった素材の使用に加え、作品の中に「3次元」をリアルタイムに登場させることを本格的に実現した、次世代のオーサリングツールです。

通常、作品を制作する作業の中では、特にアニメーションを制作する作業に非常に多くの時間を必要としました。1枚1枚の絵のデータを、長い時間をかけて作成する必要があるからです。さらに絵のデータというのは、アニメーションする枚数を増やしたり、1枚1枚のサイズやクオリティを上げるほど、非常に大容量のデータとなってしまいます。

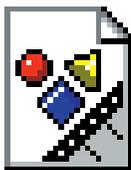
DigitalLocaでは、そういったマイナス部分を「リアルタイム3D」の概念を用いることによって減少させました。3次元の描画画面を、コンピュータにリアルタイムに計算させることにより、作品制作の編集作業を効率よく、また作品データの容量も小さくすることを可能としました。

また、カメラアングルや光源などといった、2次元にはない要素を作品の中に取り入れることができますので、新しい分野を開拓するクリエイターの方々の、新たな開発ツールとして活用して頂くことができますでしょう。

DigitalLocaを使用して、新たなマルチメディア・インタラクティブ作品を作り上げてください。

DigitalLoca 基本用語

DigitalLoca には、作業をする上で必要となるいくつかの用語がございます。はじめにここで紹介させていただきます。



プロジェクトファイル

「プロジェクト」

作品全体のデータのことをいいます。「プロジェクト」を保存すると、全体を1つのファイルとして書き出します。そのファイルの事を「プロジェクトファイル」といいます。

「3次元ワールド」

「絵」のデータというのは、奥行きがなく、縦と横にしか情報がありません。

3次元は、この「絵」に奥行きが追加されたものといえます。

それぞれ順に、横を「X」方向、縦を「Y」方向、奥行きを「Z」方向と呼びます。

DigitalLoca には、こういった仮想3次元ワールドが存在します。現実世界と同じく、「高さ」や「幅」、「奥行き」があり、この中に3次元データを配置して作品を制作していきます。

「モデリング」

3次元形状を作成する作業のことをいいます。1つ1つのモデルを組み合わせることで、さまざまな3次元モデルを作成します。

「テクスチャ」

「モデル」の表面に張り付ける「絵」のデータのことをいいます。

DigitalLoca の中では、その素材データを「テクスチャキャスト」と呼びます。

「キャスト」

作品の材料となる個々の素材データを「キャスト」と呼びます。DigitalLoca で使用するキャストには、「モデル」、「絵」、「文字」、「音」等といった 12 種類のものがあります。

「アニメーション」

素材に動きを与えることをいいます。

DigitalLoca は、「3次元モデル」や「絵」のデータなど、登録した素材にアニメーションを付けて作品を制作することができます。

作業に入る前に

チュートリアルガイドを進めるにあたって、セルシスのウェブサイトからサンプルデータ「Tutorial.zip」をダウンロードして解凍しておいてください。保存先は、DigitalLocaがインストールされているフォルダ内をお勧めします。以下は、保存先のパス例です。

— c:\Program Files\DigitalLoca 3.0\
DigitalLoca3 for Stylos サンプル

サンプル素材を使用して説明を行う場合がありますので、通常のインストールを済ませた状態のコンピュータで作業を行ってください。



DigitalLoca
アイコン

作業に入る前に、以下の点に注意してください。

- DirectX はインストールされていますか。
- モニターの表示は 16Bit 以上のカラーモードに設定されていますか。
- サウンドカードは使用可能ですか。

上記の準備が整ったら、さっそくアプリケーションを立ち上げてみましょう！

はじめに、素材を用意する作業から行います。

●本チュートリアルには、1部に旧第2版の図版画像が使用されていますが、操作や学習方法は同じものです。

第 1 章 素材の作成

1 章では

第 1 章では、プロジェクトの中に素材となるデータを用意する作業を中心に学習します。

■この章で学習する内容

- モデルキャストを作成する
- モデルキャストの材質を設定する
- モデルキャストの形状を編集する
- 素材を 3 次元ワールドに配置する

なお、チュートリアルで使用する素材は、セルシスのウェブサイトを用意されているサンプルデータ「DigitalLoca3 for Stylos サンプル」の中に入っています。予めダウンロードをして、圧縮ファイルを解凍しておいてください。

モデルキャストを作成する

はじめに、3次元データの「モデルキャスト」を作成してみましょう。一番簡単な作成方法は、ツールボックスの「プリミティブ」のタブ内から形状を呼び出す方法です。



ツールボックス「プリミティブ」

プリミティブでは、予め DigitalLoca の中に用意されている素材を、ボタン1つで呼び出すことができるウィンドウです。各ボタンには、呼び出す素材の絵が表示されています。今回は手始めに「球体」モデルを呼び出してみましょう。

■プリミティブウィンドウからモデルを呼び出す



球体モデル呼び出し

次の手順を行ってください。

- 1) プリミティブウィンドウの球体モデル呼び出しボタンをマウスでクリックします。
- 2) 形状の詳細を設定するダイアログボックスが表示されますが、今回はそのまま「OK」ボタンをクリックしてください。

プロジェクトの中に「球0」モデルが1つ作成されました。

ここで、画面の中の「キャスト」と書かれたウィンドウを見てください。モデルキャストリストの中に、「球0」というキャストが追加されました。このように、プロジェクトの中に作成されたキャストは、この「キャストウィンドウ」のリストに名前が登録されます。

このキャストウィンドウは、プロジェクトの中に作成された全キャストを管理するウィンドウです。

ここで全てのキャストタブを表示するように、キャストウィンドウ上でマウスの右ボタンをクリックし、表示されるウィンドウから「タブの多列表示」を選択してください。

キャストウィンドウは、種類ごとに分けられたリストに、作成したキャスト名が追加されていきます。このほかにも、ここからキャストを作成したり、キャストを編集するウィンドウを開いたりする機能を持っています。



キャストウィンドウ

モデルキャストの質感を設定する

それでは、今度はこの球体モデルキャストの色や質感を設定してみましょう。モデルキャストには、色や質感を付けたり、絵を貼り付けるといった編集を行う事ができます。

モデルキャストを編集するウィンドウを開いてみましょう。

■モデルのプロパティウィンドウを表示する

次の手順を行ってください。

- 1) キャストウィンドウのモデルキャストリストの中の、「球0」の名前をマウスでダブルクリックします。

画面に、モデルを編集するウィンドウが表示されました。

このキャストを編集する作業ウィンドウのことを、「モデルプロパティウィンドウ」と呼びます。もう1つのウィンドウには、モデルの形状が表示されています。このウィンドウは「モデルビューウィンドウ」と呼びます。



モデルビューウィンドウ

ここで、モデルビューウィンドウの機能を一部紹介します。
3つの操作をご紹介しますので、それぞれ簡単に試してみてください。
ださい。

- 表示されているモデルを別の角度から見る場合は、モデルビューウィンドウでマウスダウンして、見たい方向へそのままドラッグします。
- 表示されているモデルを近くや遠くから見る場合は、キーボードの [Ctrl]+[Shift] を押しながら、モデルビューウィンドウの画面でマウスダウンして、上下方向にそのままドラッグします。
- 表示されているモデルを上下左右にスライドして見る場合は、キーボードの [Shift] を押しながら、モデルビューウィンドウの画面でマウスダウンして、スライドしたい方向にそのままドラッグします。

次からの作業中に、モデルビューウィンドウで形状を確認する際には、これらの操作を行ってみてください。

色と質感を付ける

それでは、モデルに色や質感を付けてみましょう。

現在、球体モデルは灰色になっています。

今回はこの色を赤色に変更してみましょう。



モデルプロパティウィンドウ

■モデルに色を付ける

次の手順を行ってください。

- 1) モデルプロパティウィンドウの、質感設定にあるモデルの色設定の灰色の四角をクリックしてください。別ウィンドウの色の選択ウィンドウが表示されます。
- 2) 色の選択ウィンドウのカラーパレット上でマウスをドラッグし赤色を選択します。または、スライダーの値を変更して色の選択を行います。
- 3) 色の選択が終わったら「OK」を押してモデルの色を決定します。

モデルビューウィンドウを見てみると、球体モデルが赤色になったのが分かります。赤色が気に入らない場合は、先ほどの手順で、違う色に変更して頂いて構いません。

では次に、ハイライトの設定を行ってみましょう。ハイライトとは強い光が当てられた時にキラリと光る明るい部分の事をいいます。ハイライトは通常、鉄のような質感や、つやのある物体に見せるモデルに設定します。

■モデルにハイライトを付ける

次の手順を行ってください。

- 1) モデルの色設定をしたのと同じように、質感設定のハイライト色設定の白色の四角をクリックしてください。別ウィンドウの色の選択ウィンドウが表示されます。
- 2) 色の選択ウィンドウのカラーパレット上でマウスをドラッグし、ハイライト色を選択します。または、スライダーの値を変更して色の選択を行います。
- 3) ハイライト色の選択が終わったら「OK」を押して色を決定します。
- 4) ハイライト指数設定のスライダーの値を変更します。
(数値が大きいほどハイライトが強くなります。)
今回は「0.7」に設定します。

これでモデルの表面に今指定した色のハイライトが表示されるようになりました。モデルビューウィンドウでモデルを回転してみると、ハイライトの状態がよく分かります。

テクスチャを貼り付ける

今度はこの球体に、「テクスチャ」と呼ばれる「絵」のデータを貼り付けてみましょう。

DigitalLoca ではこの素材を「テクスチャキャスト」といいます。

まずはじめに、貼り付ける「テクスチャキャスト」をプロジェクトの中に用意する必要があります。

プロジェクトの中に「テクスチャキャスト」を作成してみましょう。材料となる「絵」のデータは、外部ファイルから読み込みます。

■テクスチャキャストを作成する

次の手順を行ってください。

- 1) キャストウィンドウのリストを、モデルキャストリストからテクスチャキャストリストに切り替えるために、キャストウィンドウ上部にある「texture」と書かれたタブをマウスでクリックします。



テクスチャキャストの切替えタブ

- 2) テクスチャキャストリストの中の、何も登録されていない欄をマウスでダブルクリックします。
- 3) 「Texture ファイルの読み込み」ダイアログボックスが表示されますので、DigitalLoca チュートリアル素材の中のディレクトリ「Tutorial¥Walker¥Image¥Maptest.bmp」を選択して読み込みます。

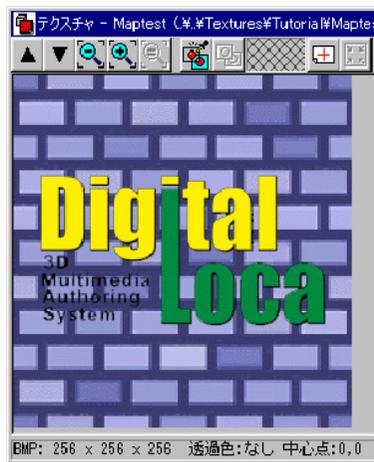
「BMP」フォーマットの絵のデータを読み込んで、テクスチャキャストが作成されました。

確認の為に、テクスチャキャストのプロパティウィンドウを開いて、読み込まれた絵を表示してみましょう。

■テクスチャのプロパティウィンドウを表示する

次の手順を行ってください。

- 1) キャストウィンドウのテクスチャキャストリストの中の、「Maptest」の名前をマウスでダブルクリックします。



テクスチャプロパティウィンドウ

画面に、テクスチャキャストを編集するウィンドウが表示されました。このウィンドウが、「テクスチャプロパティウィンドウ」です。

テクスチャプロパティウィンドウには、抜け色の指定を設定する機能があります。

今回使用するテクスチャキャストには、抜け色を指定する必要がありませんので、そのままモデルキャストに貼り付けます。

■モデルにテクスチャをマッピングする

次の手順を行ってください。

- 1) モデルキャスト「球0」のモデルビューウィンドウを表示します。

- 2) キャストウィンドウの「texture」タブをマウスでクリックしてテクスチャキャストリストに切り替えます。テクスチャキャストリストの「Maptest」の名前でマウスダウンします。
- 3) そのまま、「球0」モデルを表示しているモデルビューウィンドウの中へドラッグ&ドロップします。

「ドラッグ&ドロップ」というのは、マウスダウンしたままカーソルをドラッグし、目的の場所でマウスボタンを離す作業のことをいいます。

これで、球体モデルにテクスチャキャストがマッピングされました。また、別の方法として、モデルプロパティウィンドウのテクスチャ1の設定の、テクスチャ設定の項目から選択してテクスチャを設定することもできます。

モデルにテクスチャをマッピングする場合、テクスチャの色がモデルの色の影響を受けないようにするために、モデル自身の色を白に近い色に変えておくといよいでしょう。

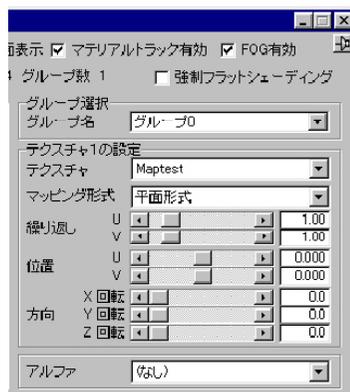
それではモデルビューウィンドウのモデルを回転させて、テクスチャの貼り付け具合を確認してみてください。

次の作業では、このテクスチャの貼り付け具合を調節する作業を行っていきます。

■貼り付け位置を移動する

次の手順を行ってください。

- 1) モデルプロパティウィンドウのテクスチャ1タブの中での、「位置」設定にある「U」「V」のスライダーを左右へ移動します。



- 2) 「方向」設定にある「X回転」「Y回転」「Z回転」のスライダーを左右へ移動します。

これらのスライダーを左右に移動することで、マッピング位置を変更することができます。

それでは初期の設定に戻しておいてください。(全て「0」です。)次にこのマッピングをタイル状に表示してみましょう。

■マッピングをタイル状にする

次の手順を行ってください。

- 1) モデルプロパティウィンドウの中のテクスチャ1タブの、「繰り返し」設定にある「U」のスライダーの値を「1」から「2」に変更します。
- 2) 繰り返し設定にある「V」のスライダーの値を「1」から「2」に変更します。

マッピングされている状態が、縦横に1つずつ増幅されて表示されました。この機能は、マッピング形式が「平面形式」の場合のみ設定することができます。

設定方法を理解したら、最後にそれぞれを元の「1」の値に戻しておいてください。

色々な角度から見て分かるように、テクスチャキャストはモデルの正面からまっすぐ貫くように貼り付けられています。

DigitalLocaでは、このマッピング形式をいくつかのタイプから選ぶことができます。

ここでマッピングの形式を変更してみましょう。

■マッピング形式を変更する

次の手順を行ってください。

- 1) モデルプロパティウィンドウの中のテクスチャ1タブの、「マッピング形式」設定の形式名を表示する欄でマウスダウンします。
- 2) 表示されるプルダウンメニューの中から、「球体形式」を選択します。

これで、球体の形状にテクスチャを貼り付けるための形式に変更されました。またマッピング形式にはいくつかの形式がありますので、ここで試してみてください。最後に、モデルプロパティウィンドウを閉じておきましょう。

ここまでで、モデルの色や質感設定、マッピングの設定を説明してまいりました。他にも設定はいくつかありますが、今回は省略いたします。



テクスチャキャストについて

画像ファイルを読み込んでテクスチャキャストを作成する際には、以下のような約束事があります。

- 読み込む画像の縦と横のピクセルサイズは、それぞれ「2、4、8、16、32、64、128、256、512」のいずれかのピクセルサイズに変更されます。
- 画像形式は、ウィンドウズビットマップ形式 (bmp) と Jpeg 形式 (jpeg、jpg) が利用できます。ウィンドウズビットマップ形式の場合、フルカラー (24bit)、256色 (8bit) の両方の色数が使用できますが、再生パフォーマンスを考慮して 256色 (8bit) に減色することをお勧めします。なお、フルカラー (24bit) 画像を読み込む場合、DigitalLoca の方で自動的に 256色 (8bit) に減色することが可能です。

モデルキャストの形状を編集する

さてここからは、実際に作品に登場させるモデルキャストを作成していきましょう。単純な「立方体」モデルを呼び出して、その形状を変更していく手順で説明を進めていきます。

はじめに、先程説明のために作成した「球0」モデルキャストは必要ありませんので、プロジェクトの中から削除しておきましょう。

■モデルキャストを削除する

次の手順を行ってください。

- 1) キャストウィンドウのモデルキャストリストの中の、「球0」の名前をマウスでクリックして選択します。
- 2) 選択された「球0」のキャスト名の上で、マウスの右ボタンを押します。
- 3) 表示されるプルダウンメニューの中から「削除」を選びます。
- 4) 確認のダイアログボックスが表示されるので、「はい」を押して実行します。

プロジェクトの中から、「球0」モデルキャストを削除しました。それでは、プリミティブウィンドウから、先程説明した方法で、今度は「立方体」モデルを呼び出してみてください。

■立方体モデルを作成する

次の手順を行ってください。

- 1) ツールボックスの「プリミティブ」タブ内から立方体モデル呼び出しボタンをマウスでクリックします。



立方体モデル呼び出し

今度はプロジェクト内に、「立方体0」というモデルキャストが追加されました。それでは、このモデルの形状を編集するウィンドウを開いてみましょう。



形状の編集ボタン

■モデリングビューウィンドウを開く

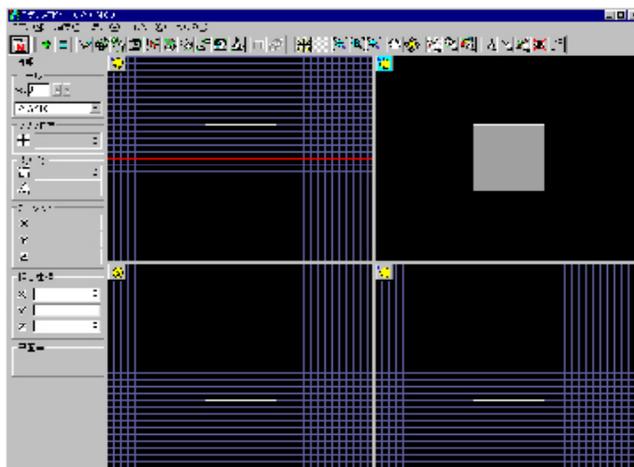
次の手順を行ってください。

- 1) 「キャストウィンドウ」のモデルキャストリストに登録された「立方体0」名前をダブルクリックして、モデルのプロパティウィンドウを開きます。
- 2) モデルプロパティウィンドウの中の、「形状の編集」ボタンをマウスでクリックします。

作業画面に、新しい作業ウィンドウが表示されました。

このモデルの形状を編集するウィンドウを、「モデリングビューウィンドウ」と呼びます。

はじめに、この「モデリングビューウィンドウ」の作業画面について簡単に説明します。



モデリングビューウィンドウ

モデリングビューウィンドウの編集画面

ウィンドウの中央には、編集するモデルの形状が表示されています。ここで表示したモデルを直接編集していきます。通常は一度に4方向から見られる設定となっています。

モデルを表示している各ウィンドウの左上には、黄色い顔のアイコンがあります。このアイコンは、モデルをどの方向から見ているかを表しています。

以下に簡単に説明します。

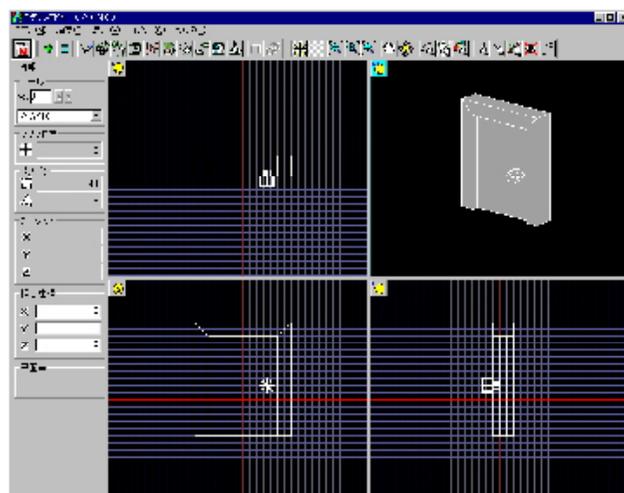


ウィンドウの上部には、モデルを編集する機能コマンドが並べられています。編集用の工具箱のようなものです。

ウィンドウの左側には、モデルのさまざまな情報を表示する窓があります。ここは、編集作業の手助けをする役割をします。

今回はこの「立方体0」モデルの形状を編集して、作品の中に登場するキャラクターの1つ、ドアモデルを作成してみます。基本的な形状の編集作業を覚えていくために、様々な方法を交えながら制作していきます。

モデリングビューウィンドウで形状を編集する



ドア完成予想図

3次元モデルというのは、通常1つ1つの「面」を組み合わせで作られています。この1つ1つの「面」のことを一般的に「ポリゴン」と呼びます。また、ポリゴンを形成しているポイントを「頂点」と呼びます。

モデリングビューウィンドウで編集するポリゴンには3角形と4角形があります。

読み込んだ「立方体0」モデルも、この「ポリゴン」という板を組み合わせで作られています。

それではさっそく「立方体0」モデルを変形して、ドアを作ってみましょう。

■頂点選択を移動する

次の手順を行ってください。

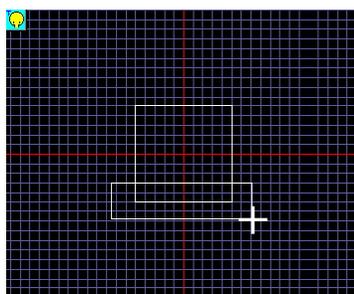
- 1) 上部にあるコマンドボタンの中から、「範囲選択」を押します。
(またはキーボードの「B」を押します。)
- 2) 「上面」から表示された視点上で、立方体の下面にある頂点を囲むようにカーソルをドラッグします。



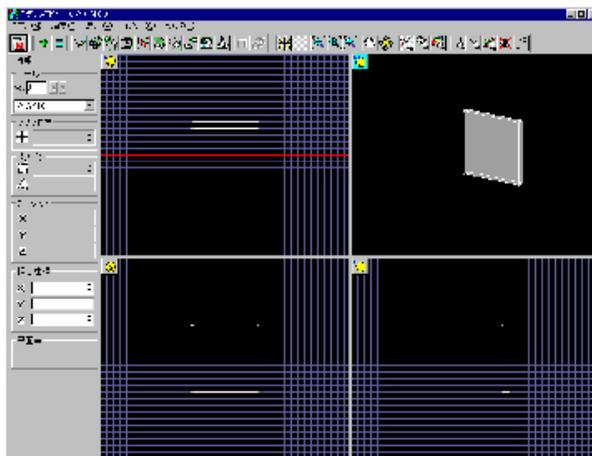
選択範囲



上面の視点



- 3) 選択した頂点を、幅が1グリッド分になる所までドラッグ移動します。(グリッドとは、パードビュー以外の視点上に網目状に表示されている目盛りのことです。)
- 4) キーボードの「スペース」を押して、頂点の選択状態を解除します。



選択した部分を移動する操作を行い、前の図のような形になりました。



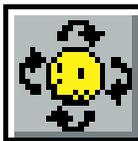
バードビュー

ここで、右上にある視点を見てください。この、顔が斜めを向いているアイコンのウィンドウを「バードビュー」といいます。バードビューは、モデルの表示方向を自由に回転できる便利な視点です。また、ポリゴン面も表示されています。バードビューを回転させるには以下の操作を行います。

■バードビューを回転させる

次の手順を行ってください。

- 1) 上部にあるコマンドボタンの中から、「バードビューのターン」を押します。(またはキーボードの「T」を押します。)
- 2) マウスカーソルをバードビューの上に移動します。
- 3) マウスダウンしてから、視点の表示を回転させたい方向にマウスをドラッグします。



バードビューのターン

このコマンドは、キーボードショートカット「T」を使用すると、非常に効率よく使用できます。また、コマンドを終了する際にはマウスの右ボタンをクリックしてください。

■バードビューを拡大・縮小する

次の手順を行ってください。

- 1) キーボードの「Ctrl」を押しながら、バードビュー上でマウスダウンし、カーソルを上下にドラッグします。

この機能も便利なので、覚えておいてください。

次は、全体を移動する作業を行います。

■全選択頂点を移動する

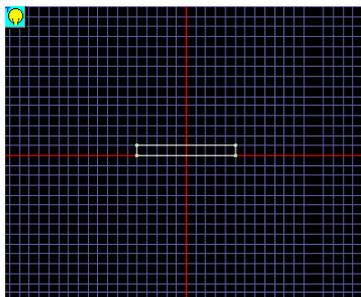
次の手順を行ってください。

- 1) キーボードの「Ctrl」+「A」を押して、全てのポリゴンの頂点を選択した状態にします。



上面の視点

- 2) 「上面」から表示された視点上でマウスダウンし、全体を中心の赤いラインの所までドラッグして移動します。(下の図を参考にしてください。)



- 3) キーボードの「スペース」を押して、頂点の選択状態を解除します。

全頂点を選択して、モデルを中心に移動させました。

今度は、モデルの上面にある頂点だけを上に移動して、縦に細長くしてみます。

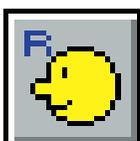
■選択頂点を移動する

次の手順を行ってください。

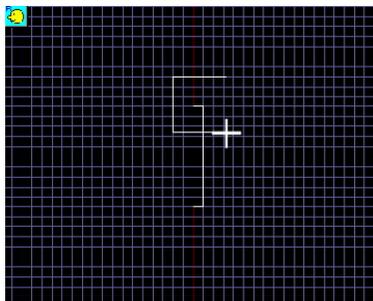
- 1) 上部にあるコマンドボタンの中から、「範囲選択」を押します。(またはキーボードの「B」を押します。)
- 2) 「右面」から表示された視点上で、モデルの上面にある頂点を囲むようにカーソルをドラッグします。



選択範囲



右面の視点



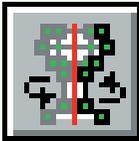
- 3) キーボードのカーソルキー「↑」を8回押して、選択した頂点を4グリッド分上に移動します。(1/2グリッドずつ移動します。)
- 4) キーボードの「スペース」を押して、頂点の選択状態を解除します。

これで、ドアの戸の部分ができあがりしました。
今度は、ドアのノブの部分を作成してみましょう。

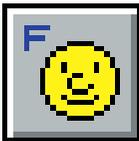
■回転体を作成する

次の手順を行ってください。

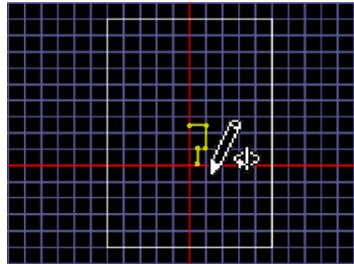
- 1) 上部にあるコマンドボタンの中から、「回転体の作成」を押します。(またはキーボードの「K」を押します。)
- 2) 「正面」から表示された視点上で、5回クリックして下の図のようなパスを作成します。(上手く形が作れなくても後から修正できますので、5回クリックしてください。)



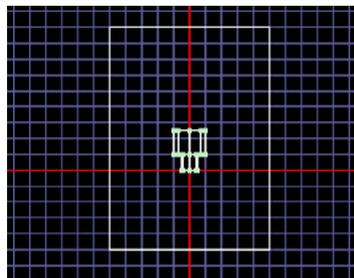
回転体の作成



正面の視点



- 3) 5回クリックしたら、キーボードの「スペース」を1回押します。(作成したパスが黄色に表示されますので、この状態で形の修正をしてください。)
- 4) もう一度、キーボードの「スペース」を押します。
※スペースキーは1回押すと形状の修正画面になり、再度押すと形状の決定になります。



回転体を作成して、ドアのノブの部分ができあがりしました。
できあがったノブの位置を移動します。



選択の回転

■選択頂点を回転する

次の手順を行ってください。

- 1) ドアノブモデルの全てのポリゴンの頂点が選択されている事を確認してください。先程の作業の続きでしたら、選択した状態になっています。上部にあるコマンドボタンの中から、「選択の回転」を押します。(またはキーボードの「L」を押します。)
- 2) モデリングビューウィンドウの左側に、回転の値を入力するパネルが表示されます。「X」の値をマウスでクリックして、「-90」と入力し、キーボードの「Enter」を押します。



選択状態の頂点が、赤いポインタを中心に、-90度X軸回転しました。そのまま作業を続けましょう。

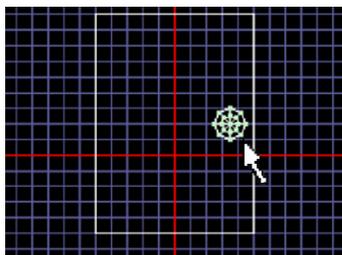
■頂点選択を移動する

次の手順を行ってください。

- 1) 先程の作業で、ドアノブモデルのポリゴンの頂点が全て選択されています。そのまま「正面」から表示された視点上でマウスダウンし、ドアのノブの位置までドラッグして移動します。(次の図を参考にしてください。また、微調整にはキーボードのカーソルキー「←→↑↓」を使用することができます。)



正面の視点



- 2) キーボードの「スペース」を押して、頂点の選択状態を解除します。



バードビュー

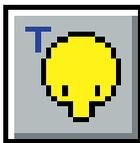
これで、ノブの付いたドアができあがりました。(バードビューを回転してモデルを確認してみてください。)

最後にドアの枠を作成します。

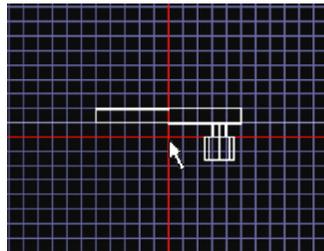
■ポリゴンを作成する

次の手順を行ってください。

- 1) 「上面」から表示された視点上で、下の図のあたりをキーボードの「Alt」を押しながらマウスの右ボタンでクリックして、中心にある赤いポイントを、1 グリッド分移動させます。



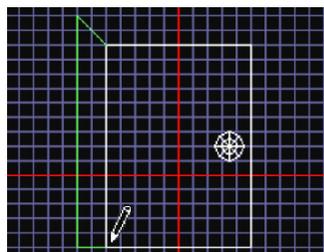
上面の視点



- 2) 上部にあるコマンドボタンの中から、「ポリゴンの追加」を押します。(またはキーボードの「A」を押します。)
- 3) 「正面」から表示された視点上で、下の図のグリッド上を、反時計周りの順番で4回クリックします。最後にキーボードの「スペース」を押します。



ポリゴンの追加



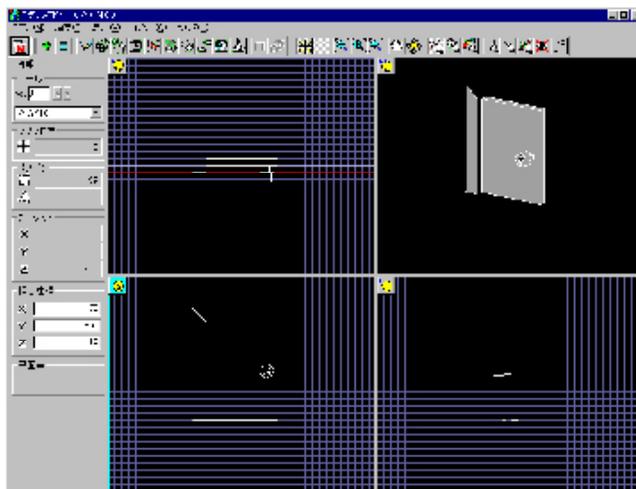
正面の視点



バードビュー

4つ頂点を作成した時点で、新しい4角形ポリゴンが下の図のように作られたと思います。

(バードビューを回転させて確認してみてください。)



次に、今作成したポリゴンを複製して、ドアの反対側にも同じポリゴンを作成してみます。

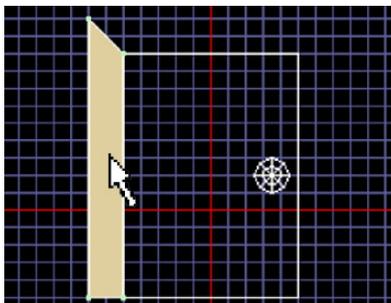


ポリゴン選択

■ポリゴンを複製してスケールを反転する

次の手順を行ってください。

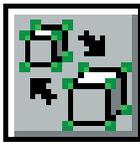
- 1) 上部にあるコマンドボタンの中から、「ポリゴン選択」を押して、頂点単位の選択モードから、ポリゴン単位の選択モードに切り替えます。(またはキーボードの「P」を押します。)
- 2) 先ほど作成したポリゴンの真ん中あたりをマウスでクリックします。ポリゴンが選択されて、黄色く表示されます。



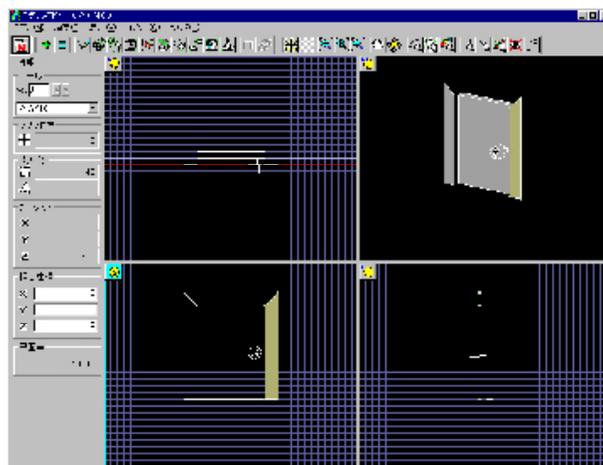
- 3) ポリゴンが選択された状態のまま、キーボードの「Ctrl」+「C」を押して、選択状態のポリゴンをコピーします。
- 4) 上部にある編集メニューの中から「ペーストモード」を選んだ後、モードを「グループ0」に変更します。



- 5) 今度はキーボードの「Ctrl」+「V」を押して、コピーしたポリゴンをペースト（貼り付け）します。
- 6) 複製したポリゴンが選択状態になっていますので、そのまま上部にあるコマンドボタンの中から「選択の拡大・縮小」を押します。（またはキーボードの「S」を押します。）
- 7) モデリングビューウィンドウの左側に、スケールの値を入力するパネルが表示されます。「X」の値をマウスでクリックして「-100」と入力し、キーボードの「Enter」を押します。



選択の拡大・縮小



- 8) 選択状態のポリゴンが、赤いポインタを中心にして、X方向に反転しました。キーボードの「スペース」を押して、ポリゴンの選択状態を解除します。

ドアの反対側にも、ポリゴンを作成することができました。しかし、反転したポリゴンを見てみると、元のポリゴンと色が異なって表示されています。これは、反転したポリゴンの面が裏を向いているためです。ポリゴンには表と裏があり、実際の再生画面上ではポリゴンの裏面は見る事ができません。モデルビューウィンドウでは表示されますが、裏は表よりも少し暗い色で表示されます。それでは、このポリゴンの表と裏を反転する作業を行いましょ

■ポリゴンの表裏を反転する

次の手順を行って下さい。

- 1) 上部にあるコマンドボタンの中から、「選択ポリゴンの表裏反転」を押します。(またはキーボードの「R」を押します。)
- 2) 「正面」から表示された視点上で、右側にあるポリゴンの真ん中あたりをクリックします。
- 3) 選択したのと同時に、ポリゴンの表と裏が反転しました。「選択ポリゴンの表裏反転」コマンドを終了するために、一度マウスの右ボタンをクリックします。
- 4) キーボードの「スペース」を押して、ポリゴンの選択状態を解除します。

ポリゴンの表裏を反転させる事ができました。バードビューを回転させて確認してみてください。

ポリゴンの反転作業について

作業の途中ですが、「選択ポリゴンの表裏反転」コマンドを使ってポリゴンを反転させる作業について、少しテクニックをご紹介します。

反転するポリゴンの数が少ない場合には、今のように1つずつクリックして反転していきます。反転しなければならないポリゴンが多い場合には、ポリゴン選択モードに切り替えて、キー



選択ポリゴンの
表裏反転



正面の視点



バードビュー

ボードの「Shift」を押しながら「範囲選択」などを使用して、反転するポリゴンを複数選択しておきます。最後に「選択ポリゴンの表裏反転」機能を選択して、ウィンドウ内のポリゴンが表示されていない場所をクリックすると、選択状態のポリゴン全てを一度に反転することができます。

次は少し変わったポリゴンの作り方を利用して、ドア枠の上の部分を作ってみましょう。

■ 選択ポリゴン以外を隠す

次の手順を行ってください。

- 1) 「正面」から表示された視点上で、ドアの枠となる左右のポリゴンそれぞれの中心あたりを、キーボードの「Shift」を押しながらクリックします。
- 2) 2つのポリゴンが選択されました。そのまま、上部にあるコマンドボタンの中から、「選択のみ表示」を押します。(またはキーボードの「H」を押します。)



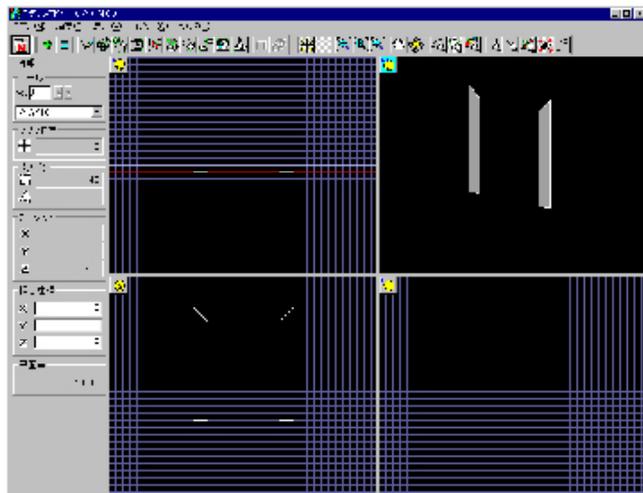
正面の視点



選択のみ表示



バードビュー



これからの作業をしやすくするために、選択していないポリゴンを隠して見えなくしました。

■新規ポリゴンの頂点同士をくっつける

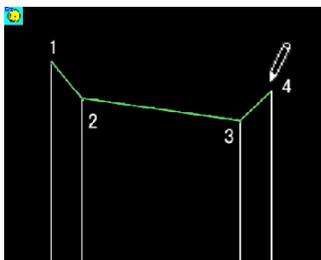
次の手順を行ってください。

- 1) バードビューの表示アングルを、次の図のような角度にします。

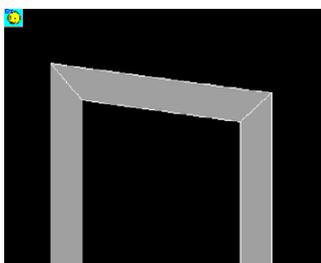


ポリゴンの追加

- 2) 上部にあるコマンドボタンの中から、「ポリゴンの追加」を押します。(またはキーボードの「A」を押します)
- 3) キーボードの「Shift」を押しながら、左のポリゴンの頂点から反時計回りに4回、下の図のようにクリックします。最後にキーボードの「スペース」を押します。



2つのポリゴンの間に新しいポリゴンが作成されました。「ポリゴンの追加」とキーボードの「Shift」を使用すると、頂点が接着された状態のポリゴンを作成することができます。



— ポリゴンの作成ヒント —



頂点選択

なお、キーボードの「Shift」を使用しない場合は、独立したポリゴンが作成されることとなります。このように任意の独立したポリゴンを作成して、後から他のポリゴンの頂点と接着させる方法もあります。その場合は、まず、上部のボタンの中から「頂点選択」を押して、頂点選択モードに切り換えます。（またはキーボードの「V」を押します。）

次に、接着したい頂点をドラッグして、対照ポリゴンの頂点に近づけてから、マウスの左ボタンを押したまま、マウスの右ボタンを押すことで頂点どうしを接着させることができます。

それでは、頂点同士がつながったポリゴンの作成作業が終わりまりましたので、隠していたポリゴンを表示してみましょう。

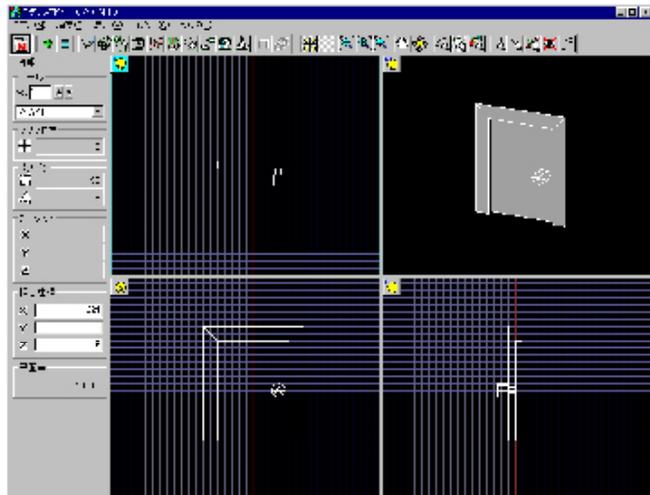
■隠していたポリゴンを表示する

次の手順を行ってください。

- 1) 上部にあるコマンドボタンの中から、凹んでいる「選択のみ表示」を押して元に戻します。（またはキーボードの「H」を押します。）



選択のみ表示



だいぶドア枠の形になってきました。しかし、これではドア枠に厚みがありません。そこで、このドア枠の形を引き伸ばし少し厚みを付けましょう。

■グループ選択をする

次の手順を行ってください。

- 1) 上部にあるコマンドボタンの中から、「ポリゴン選択」を押して、頂点単位の選択モードから、ポリゴン単位の選択モードに切り替えます。(またはキーボードの「P」を押します。)
- 2) 「正面」から表示された視点上(またはバードビュー上)で、ドアの枠となる3つのポリゴンの内、どれか1つをクリックして選択します。
- 3) 上部にあるコマンドボタンの中から、「共通頂点パーツの選択」を押します。(またはキーボードの「G」を押します。)

選択したポリゴンと、頂点が間接的にくっついているポリゴン全てが選択されました。



ポリゴン選択



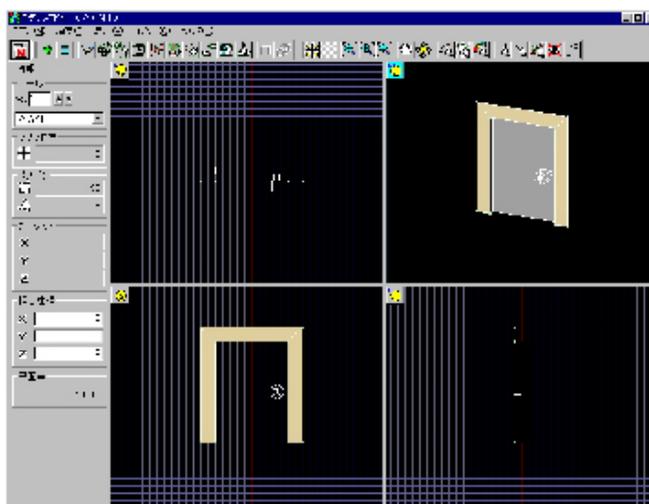
正面の視点



バードビュー



共通頂点パーツの
選択

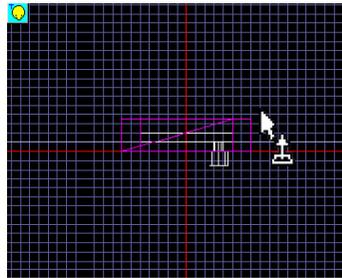


■選択ポリゴンを引き伸ばす

次の手順を行ってください。

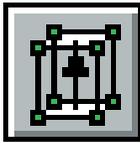
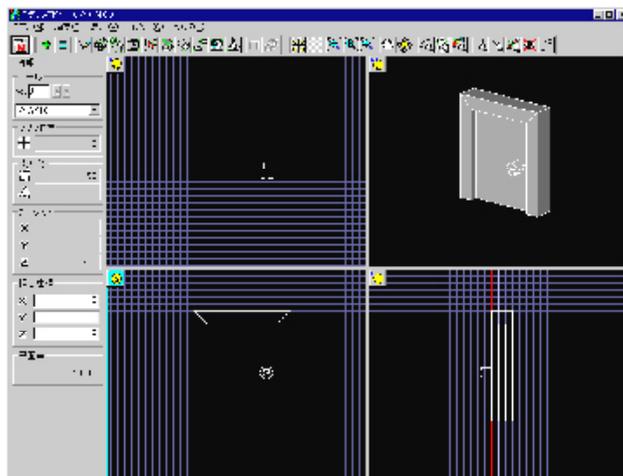
1) 先ほどの作業で、ドア枠となるポリゴン全てが選択されています。そのまま、上部にあるコマンドボタンの中から、「選択の引き伸ばし」コマンドを押します。(またはキーボードの「E」を押します。)

2) 「上面」から表示された視点上の、ポリゴンが表示されていない所でマウスダウンします。そのまま、まっすぐ上に引き伸ばすようにドラッグします。

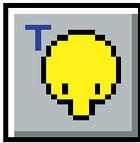


3) 上の図のように、表示されているグリッドの3つ先の所までドラッグしたら、マウスの左ボタンを離します。
4) キーボードの「スペース」を押して、ポリゴンの選択状態を解除します。

ドア枠がほぼ完成しました。しかし、ポリゴンを引き伸ばす際に、表裏が反転したポリゴンが作られています。



選択の引き伸ばし



上面の視点



ポリゴン選択



バードビュー



共通頂点パーツの
選択



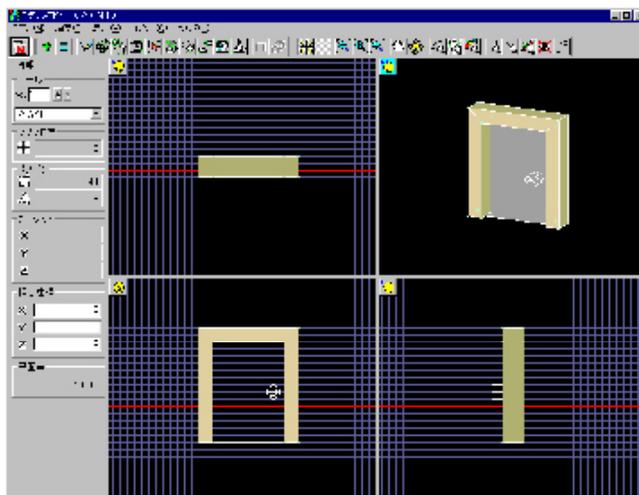
選択のみ表示

■グループ選択をする

次の手順を行ってください。

- 1) もう一度、上部にあるコマンドボタンの中から、「ポリゴン選択」を押して、頂点単位の選択モードから、ポリゴン単位の選択モードに切り替えます。(またはキーボードの「P」を押します。)
- 2) 「正面」から表示された視点上(またはバードビュー上)で、ドアの枠となるのポリゴンの内、どれか1つをクリックして選択します。
- 3) 上部にあるコマンドボタンの中から、「共通頂点パーツの選択」を押します。(またはキーボードの「G」を押します。)

引き伸ばされたポリゴンを含み、頂点が間接的にくっついているドア枠の全てが選択されました。

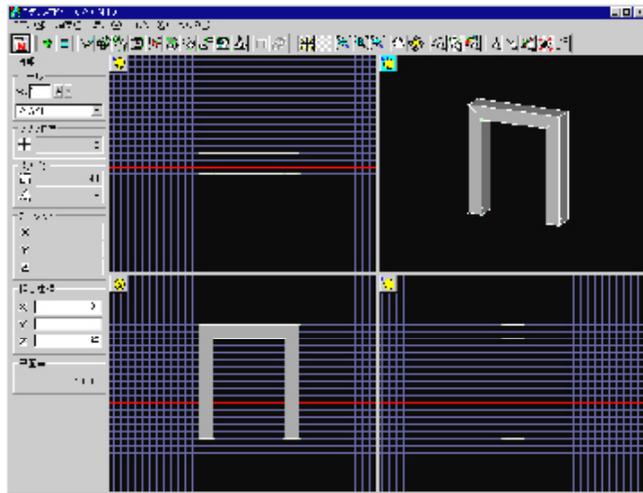


■選択ポリゴン以外を隠す

次の手順を行ってください。

- 1) ドア枠の全てのポリゴンが選択されたまま、上部にあるコマンドボタンの中から、「選択のみ表示」を押します。(またはキーボードの「H」を押します。)

ドア枠のポリゴンの表裏を統一するために、選択していないポリゴンを隠して見えなくしました。



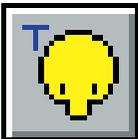
■ 選択状態の反転

次の手順を行ってください。

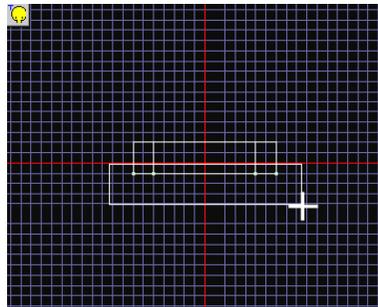
- 1) 上部にあるコマンドボタンの中から、「範囲選択」を押します。
(またはキーボードの「B」を押します。)
- 2) 「上面」から表示された視点上で、ドア枠の下面にある頂点を囲むようにカーソルをドラッグします。



選択範囲

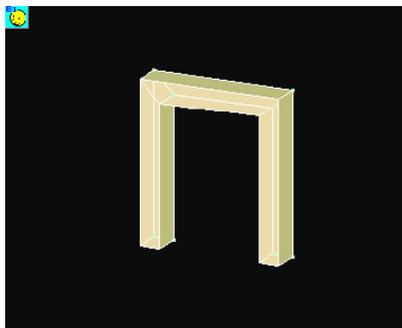


上面の視点



- 3) 「バードビュー」で、ドア枠の正面のポリゴンが選択されたことを確認します。

- 4) 上部にある編集メニューの中から「選択状態の反転」を選びます。(またはキーボードの「Ctrl」を押したまま、「R」を押します。)



これで、ポリゴンの表裏を反転させたいドア枠の後ろ部分のみが選択状態になりました。

■選択ポリゴンの表裏反転

次の手順を行ってください。

- 1) ドア枠の後ろ部分のポリゴンが選択されたまま、上部にあるコマンドボタンの中から、「選択ポリゴンの表裏反転」を押します。(またはキーボードの「R」を押します。)
- 2) 「バードビュー」から表示された視点上で、ポリゴン以外の任意の部分をクリックします。



選択ポリゴンの
表裏反転



バードビュー



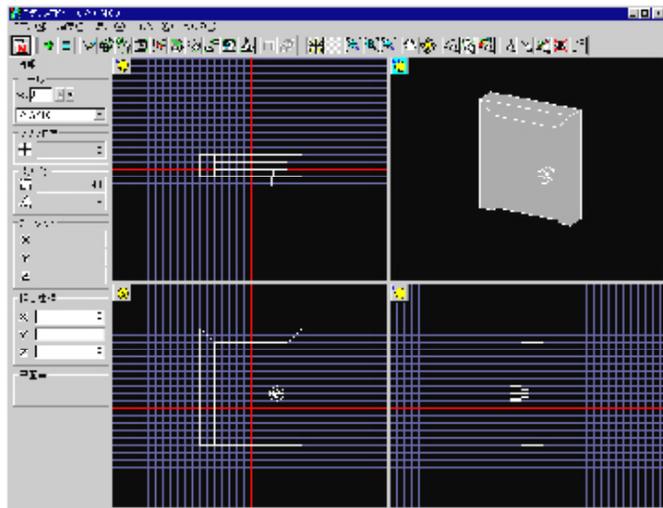
- 3) ドア枠後ろ部分のポリゴンの表と裏が反転しました。「選択ポリゴンの表裏反転」コマンドを終了するために、一度マウスの右ボタンをクリックします。



選択のみ表示

- 4) キーボードの「スペース」を押して、ポリゴンの選択状態を解除します。
- 5) 上部にあるコマンドボタンの中から、凹んでいる「選択のみ表示」を押して元に戻します。(またはキーボードの「H」を押します。)

ドア枠全てのポリゴンの表裏を統一させる事ができました。バードビューを回転させてモデル全体を確認してみてください。

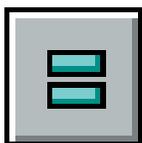


■スムージングによる共有頂点の結合

次の手順を行ってください。

- 1) 上部にあるコマンドボタンの中から、「スムージング角による頂点の結合」を押します。(またはキーボードの「I」を押します。)
- 2) 「スムージング角による頂点の結合」の編集画面が開かれます。そのまま「OK」ボタンを押します。

「スムージング角による頂点の結合」は、ポリゴン編集によって新しく作成された頂点と既存の頂点間とのつながりを結合させます。



モデルキャストの更新

■モデルキャストを更新する

次の手順を行ってください。

- 1) 上部にあるコマンドボタンの中から、「モデルキャストの更新」を押します。(またはキーボードの「Ctrl」+「S」を押します。)

これで、立方体モデルキャストの形状は、ドアの形状に更新されました。ここで一度モデリングビューウィンドウを終了し、モデルのプロパティウィンドウに戻ってみましょう。モデリングビューウィンドウを閉じてください。



立方体だったモデルが、作成したドアの形状に変更されています。(モデルビューウィンドウのモデルを回転して、形を確認してみてください。)

ここで「プロジェクト」全体を、ファイルに保存しておきましょう。

■プロジェクトを保存する

次の手順を行ってください。

- 1) メニューの「プロジェクト」から「プロジェクトに名前を付けて保存」を選びます。
- 2) ファイルを保存する先を選び、ファイル名を「Door」と入力して「保存」を押します。

「Door.lcp」という名前のファイルが保存されました。次回からは、メニューの「プロジェクト」から「プロジェクトの上書

き保存」を実行することで、更新内容がこのファイルに上書きされます。

モデルをキャストごとに分割する

作成したドアモデルは、一かたまりのモデルキャストです。このままでは、パーツごとに別々のアニメーションを設定する事ができません。

そこで、このモデルキャストをパーツごとに分割して、別々のモデルキャストにする作業を行います。

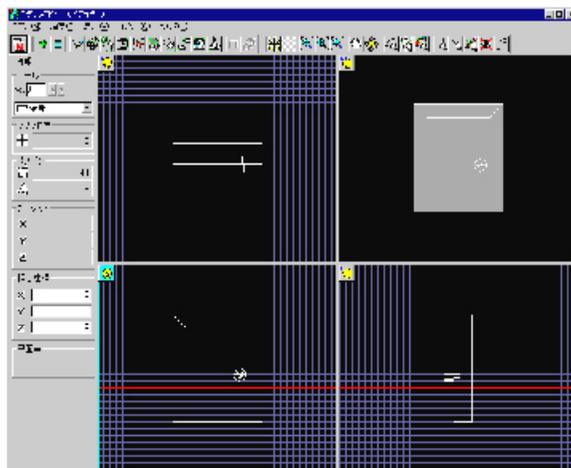
まずはじめに、先ほど作成したドアモデルのキャスト名を変更しておきましょう。

■キャスト名を変更する

次の手順を行ってください。

- 1) キャストウィンドウのリストの中の、「立方体 0」の名前を選択して、マウスで右クリックします。
- 2) メニューが表示されるので「名前の変更」をクリックします。
リスト名が反転表示になり、入力状態になります。「ドア全部」と入力し、キーボードの「Enter」を押します。

モデルキャスト名が、「立方体 0」という名前から「ドア全部」という名前に変更されました。それでは再度モデリングビューウィンドウを開いて、モデルを分割する作業に入ります。先程説明した手順で、「ドア全部」モデルのモデリングビューウィンドウを表示してください。



この「ドア全部」モデルを、「ドア枠」、「ドア（ドアノブ）」の2つのパーツに分けてみます。

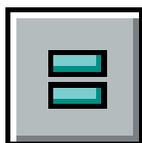
■「ドア枠」となるポリゴンを別キャストにする

次の手順を行ってください。

- 1) 上部にあるコマンドボタンの中から、「ポリゴン選択」を押して、頂点単位の選択モードから、ポリゴン単位の選択モードに切り替えます。(またはキーボードの「P」を押します。)
- 2) 「ドア枠」となるポリゴンのどれかをマウスでクリックします。ポリゴンが選択されて、黄色く表示されます。
- 3) 上部にあるコマンドボタンの中から、「共通頂点パーツの選択」を押します。(またはキーボードの「G」を押します。)
- 4) ドア枠となる全てのポリゴンが選択されました。ポリゴンが選択された状態のまま、キーボードの「Ctrl」+「C」を押して、選択状態のポリゴンをコピーします。
- 5) 上部にあるコマンドボタンの中から、「モデルキャストの新規作成」を押します。(またはキーボードの「Ctrl」+「N」を押します。)
- 6) 新規モデルキャストの編集画面が開かれます。開かれた編集画面のいずれかのウィンドウ上でキーボードの「Ctrl」+「V」を押して、コピーされたポリゴンをペースト（貼り付け）します。
- 7) 上部にあるコマンドボタンの中から、「モデルキャストの更新」を押します。(またはキーボードの「Ctrl」+「S」を押します。)
- 8) キーボードの「スペース」を押して、ポリゴンの選択状態を解除します。



モデルキャストの
新規作成



モデルキャストの更新

「ドア全部」モデルのドア枠部分のポリゴンだけを、新規モデルキャストに複製する作業を行いました。さっそくこのモデルに名前を付けておきましょう。

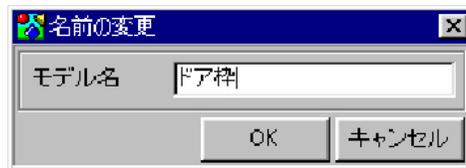
モデルビューウィンドウの左側にある窓を見てください。現在、このモデルには「モデル0」という名前が付けられています。この名前をモデリングビューウィンドウ上で「ドア枠」という名前に変更しておきましょう。



■モデリングビューウィンドウ上でキャスト名を変更する

次の手順を行ってください。

- 1) メニューの「モデル」から「モデルキャスト名の変更」を選びます。(またはキーボードの「F8」を押します。)
- 2) 名前の変更ダイアログボックスが開かれます。「ドア枠」と入力して「OK」を押します。



ドア枠だけのモデルキャストができました。モデル名が表示されている場所にある上矢印をクリックして、モデル表示を「ドア全部」モデルの表示に戻してください。



それでは、今説明した別モデルキャストに複製する手順を参考にして、残りの「ドア（ドアノブ）」モデルを別のモデルキャストにする作業を行ってみましょう。

■「ドア」となるポリゴンを別キャストにする

次の手順を行ってください。

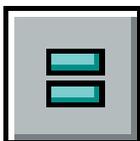
- 1) ドアの扉のポリゴンを選択し、キーボードの「Shift」を押しながらノブの部分のポリゴンを選択します。上部にあるコマンドボタンの中から、「共通頂点パーツの選択」を押します。(またはキーボードの「G」を押します。)
- 2) ドアの扉とノブのすべてが選択された状態になります。



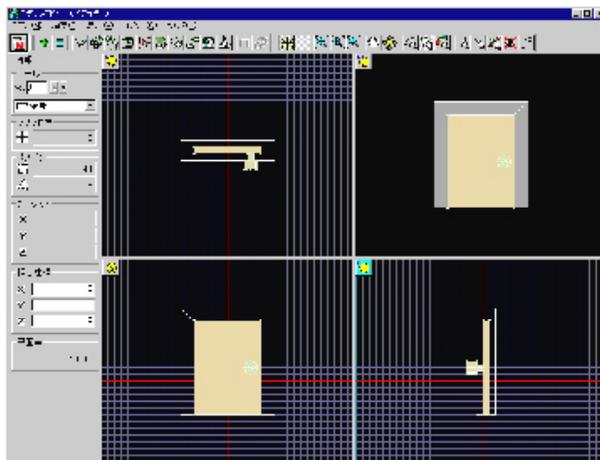
共通頂点パーツの
選択



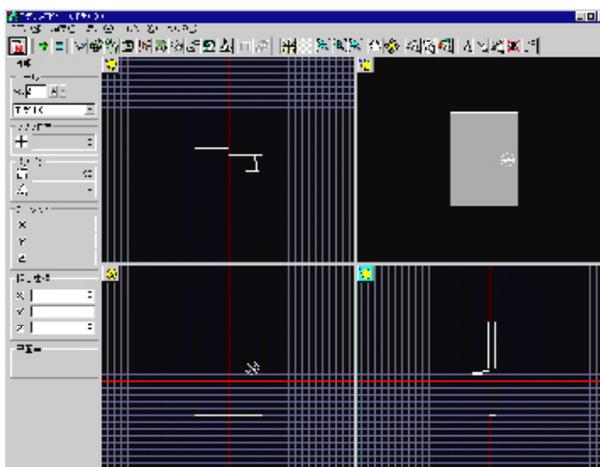
モデルキャストの
新規作成



モデルキャストの更新



- 3) 「ドア」となる全てのポリゴンが選択された状態のまま、キーボードの「Ctrl」+「C」を押して、選択状態のポリゴンをコピーします。
- 4) 上部にあるコマンドボタンの中から、「モデルキャストの新規作成」を押します。(またはキーボードの「Ctrl」+「N」を押します。)
- 5) 新規モデルキャストの編集画面が開かれます。キーボードの「Ctrl」+「V」を押して、コピーされたポリゴンをペースト(貼り付け)します。
- 6) 上部にあるコマンドボタンの中から、「モデルキャストの更新」を押します。(またはキーボードの「Ctrl」+「S」を押します。)
- 7) キーボードの「スペース」を押して、ポリゴンの選択状態を解除します。



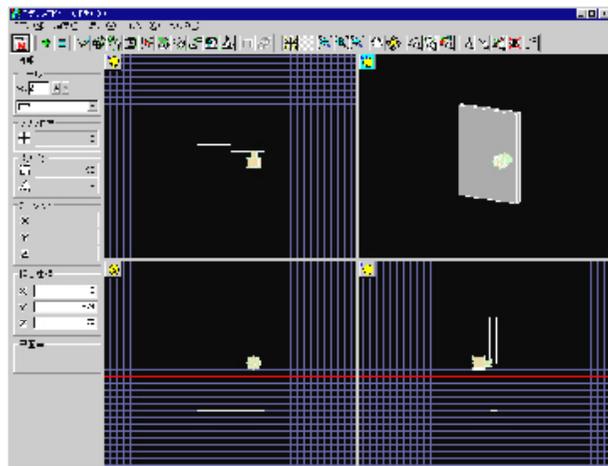
ドアの開く部分のポリゴンだけを、新規モデルキャストにする作業を行いました。このモデルには「ドア」という名前を付けておいてください。

ドアだけのモデルキャストができました。

このモデルには、後から行うテクスチャ設定の作業の為に、グループ分けの設定を行います。

1つのモデルでも「グループ」単位で分割することによって、別々の色や質感を設定することができます。ドアの扉の部分と、ノブの部分に別々の質感設定を行ってみましょう。

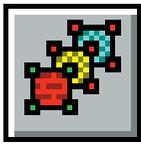
■ 「ドアノブ」となるポリゴンを別グループにする



次の手順を行ってください。

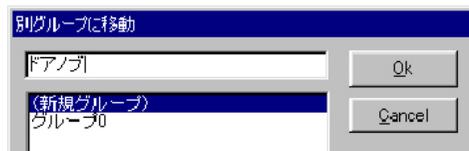
- 1) ドアのノブのポリゴンをすべて選択した状態にします。
(前の図を参考にしてください。)
- 2) モデリングビューウィンドウ上部にある「グループ編集モードに切り換え」ボタンを押し、グループ分けを行うモードに切り替えます。(または、キーボードの「Alt」を押しながら「Z」を押します。)





選択の別グループ移動

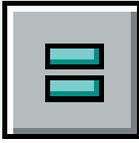
- 3) 上部にあるコマンドボタンの中から、「選択の別グループ移動」を押します。(またはキーボードの「F3」を押します。)
- 4) 「別グループに移動」ダイアログボックスが表示されますので、リストの中の「新規グループ」をクリックし、ウィンドウ上部に表示された「(新規グループ)」の名を「ドアノブ」に書き換えます。(最後に「Enter」を押してください。)



選択していたドアノブのポリゴンが、「ドアノブ」のグループに登録されました。

モデリングビューウィンドウ左側にある、「設定」タブの中のグループ名をクリックすると、「ドアノブ」というグループが追加されていることが確認できます。





モデルキャストの更新

モデルの中心を変更する

モデル編集モードに戻る前に、上部にあるコマンドボタンの中から、「モデルキャストの更新」を押します。（またはキーボードの「Ctrl」+「S」を押します。）その後で、モデリングビューウィンドウ上部にある「形状編集モードに切り換え」ボタンを押して、グループ分けを行うモードから、形状を編集するモードに戻してください。

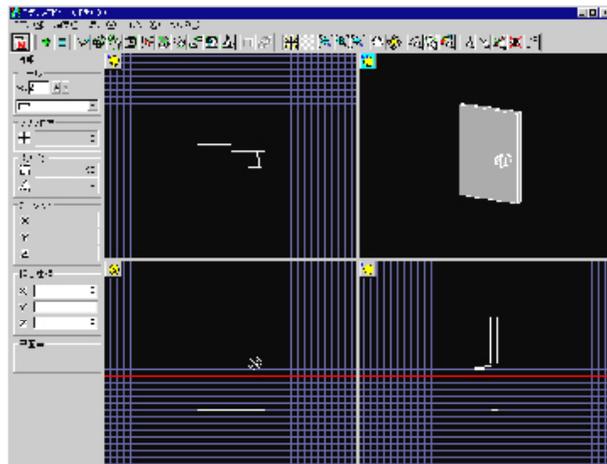


モデリングビューウィンドウの表示を、「ドア」モデルの表示にしてみましょう。

それではここでモデルの中心点を移動しておきます。

■モデルの中心点を変更する

モデルの中心点とは、ワークビューウィンドウ上で「回転値」や「スケール値」の設定を行う際に中心となる部分のことをいいます。



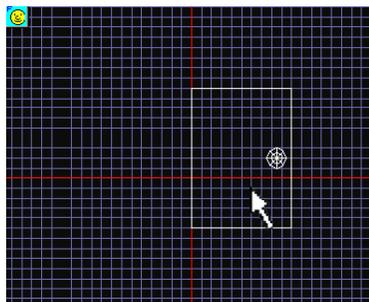
この「ドア」モデルの場合の回転軸は、ドアの中心ではなく、蝶番（ちょうつがい）の付いている端の部分になります。また、このモデルのノブの位置は、「正面」から表示された視点上で見て右側に付いていますので、このドアの回転軸は左端になります。

それでは中心点を移動しておきましょう。

■選択頂点を移動する

次の手順を行ってください。

- 1) キーボードの「Ctrl」+「A」を押して、ドアモデルのポリゴンの全頂点を選択した状態にします。
- 2) 「正面」から表示された視点上でマウスダウンします。そのまま右へドラッグして、ドアモデルの左端を中心にある赤色のグリッド線に合わせます。



- 3) キーボードの「スペース」を押して、ポリゴンの選択状態を解除します。

これで、「ドア」モデルキャストの回転アニメーションの軸を、左端にすることができました。変更を加えた箇所を更新するために、上部にあるコマンドボタンの中から「モデルキャストの更新」を押してください。（またはキーボードの「Ctrl」+「S」を押します。）

お疲れさまでした！モデリングビューウィンドウを使用した編集作業はこれで終わりです。

モデリングビューウィンドウを閉じてください。

ここまでの作業で、新しいモデルキャストの「ドア全部」、「ドア枠」、「ドア」を作成することができました。

ドアモデルを作成する方法を例にして、モデルキャストを作成する方法を説明してまいりました。これでモデリングビューウィンドウの中で使用する機能も、少しお分かり頂けたかと思えます。ウィンドウの中で使用する機能も、少しお分かり頂けたかと思えます。



各キャストに素材を設定する

先程完成させた、「ドア枠」、「ドア」、「ドアノブ」にも、色や質感を設定しておきましょう。まず、それぞれのモデルにマッピングするテクスチャキャストを準備します。

■テクスチャキャストを準備する

次の手順を行ってください。

- 1) DigitalLoca がインストールされたディレクトリの「Walker¥Image¥Door.bmp」を「新規テクスチャ」キャストとして読み込んでください。

テクスチャキャストの「Door」が作成されました。

(キャスト名はそのままで結構です。)

それでは、モデルのプロパティウィンドウを開いて、「ドア枠」、「ドア（ドアノブ）」の質感を設定しましょう。

■「ドア枠」モデルの質感設定

次の手順を行ってください。

- 1) 「ドア枠」モデルキャストのプロパティウィンドウの「モデルの色」の値を、赤 (R) = 「80」、緑 (G) = 「80」、青 (B) = 「180」、に設定します。

■ 「ドア」モデルの質感設定

次の手順を行ってください。

- 1) 「ドア」モデルキャストのプロパティウィンドウの「モデルの色」の値を、赤 (R) = 「255」、緑 (G) = 「255」、青 (B) = 「255」、に設定します。
- 2) テクスチャキャストリストの「Door」を、「モデルビューウィンドウ」にドラッグ&ドロップしてマッピングします。



■ 「ドアノブ」モデルの質感設定

次の手順を行ってください。

- 1) 「ドア」モデルキャストのプロパティウィンドウの「グループ名」をクリックし、「グループ0」から「ドアノブ」の表示に切り替えます。



- 2) プロパティウィンドウの「モデルの色」の値を、赤 (R) = 「200」、緑 (G) = 「200」、青 (B) = 「80」、に設定します。
- 3) プロパティウィンドウの「ハイライト色」の値を、赤 (R) = 「255」、緑 (G) = 「255」、青 (B) = 「180」、に設定します。

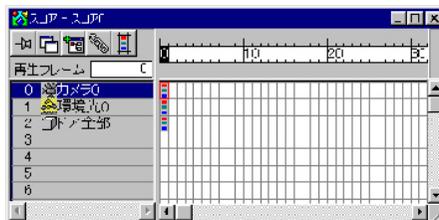
- 4) プロパティウィンドウの「ハイライト指数」の値を、「0.5」に設定します。

モデルの質感設定も終了しました。

(質感設定の説明が終わりましたので、テクスチャキャストの名前は変更して頂いて構いません。)

モデルキャストを配置する

ここからは、先程作成したドアモデルの各パーツとなるモデルキャストを、それぞれ3次元ワールドに配置して組み立てる作業を説明します。モデルキャストを3次元ワールドに登場させるには、「スコア」と呼ばれるウィンドウに登録する必要があります。



スコアウィンドウ

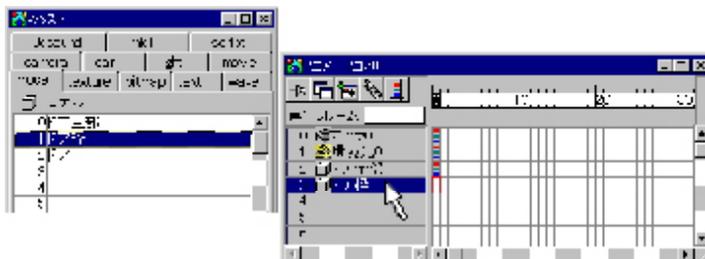
それでは先程の3つのモデルキャストを、この「スコアウィンドウ」に登録してみましょう。

(「ドア全部」モデルはすでにスコアに登録されていますので、そのまま結構です。)

■モデルキャストをスコアに登録する

次の手順を行ってください。

- 1) キャストウィンドウのモデルキャストリストの中の、「ドア枠」のキャスト名をマウスでクリックして選択します。
- 2) 選択した「ドア枠」のキャスト名でマウスダウンします。そのままカーソルをドラッグして、下の図の場所に移動し、マウスの左ボタンを離します。



「ドア枠」モデルキャストが登録されました。このキャストを登録したスコアウィンドウの横列を、「トラック」と呼びます。ここで、「スコアウィンドウ」について説明します。スコアウィンドウのトラックに登録された各キャストは、キャスト名の右側にならぶ各ブロックに、アニメーション情報等を登録していきます。この各ブロックを「フレーム」と呼びます。

それでは、先程説明した手順を参考にして、残りの「ドア（ドアノブ）」モデルキャストを、新規トラックに登録する作業を行っててください。



作業を終えると、スコア上には上の図のようにキャストが登録されています。次は、この2つのキャストを3次元ワールドに表示するために、スコアウィンドウの0フレーム目に「キーフレーム」を作成してみましょう。

■キーフレームを作成する

次の手順を行ってください。

- 1) 下の図を参考に、スコアウィンドウの2つのモデルトラックの0フレーム目を、マウスダウンしながら、カーソルをドラッグします。



- 2) 2つ目のフレームまできたら、マウスの左ボタンを離します。カーソルが手の形である事を確認し、マウスを右クリックします。表示されるプルダウンメニューから、「キーフレームを設定/全て」を実行します。

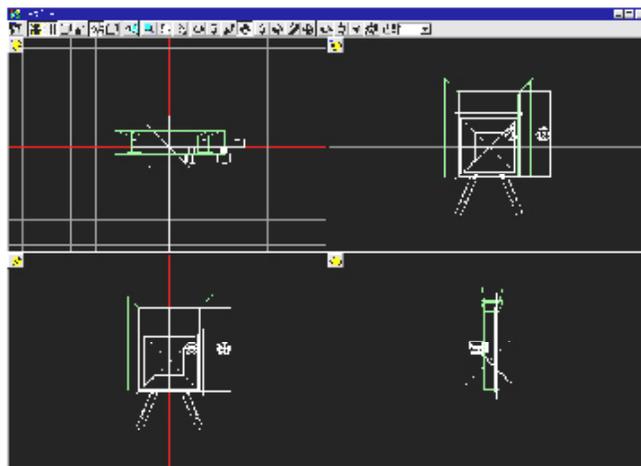


- 3) 選択した2つのフレームに、「キーフレーム」が登録されたら、他のフレームをクリックし決定します。これで、キーフレームが設定されました。



キーフレームとは、アニメーションを行うために必要な情報が設定されたフレームのことをいいます。今の作業で、3つのモデルトラックの0フレーム目に、「座標値」、「回転値」、「スケール値」、「不可視」の情報を設定しました。

また、これらのキャストの3次元ワールドでの様子を表示するのが、「ワークビューウィンドウ」です。



ワークビューウィンドウ

「ワークビューウィンドウ」は、DigitalLocaの3次元ワールドを、さまざまな視点から表示することができます。モデルなどの3次元ワールド内での配置や状態を見ることができます。

「モデリングビューウィンドウ」を、小道具の作成ウィンドウとすれば、「ワークビューウィンドウ」は、モデル、カメラ、ライト等を配置する、撮影スタジオといえるでしょう。

3次元ワールドを分かりやすくするために、初期設定では、3次元ワールドの中心から、網目状の「グリッド」が表示されています。

3次元ワールドを表示している各視点の左上には、黄色い顔のアイコンがあります。このアイコンはモデリングビューウィンドウ同様、どの方向からの視点かを表しています。

ウィンドウの上部には、3次元ワールドに配置したキャストを操作する、機能コマンドが並べられています。

モデルの組み立て作業の途中ですが、ここで、ワークビューウィンドウの操作をいくつかご紹介します。モデリングビューウィンドウと似ていますが、視点の操作方法などが多少異なります。それぞれ簡単に試してみてください。

- 各視点を上下左右にスライドさせたい時
キーボードの [Shift] を押しながら、操作する視点の上でマウスダウンして、スライドしたい方向にドラッグします。
- 各視点を前進、後退させたい時
キーボードの「Shift」 + 「Ctrl」を押しながら、操作する視点の上でマウスダウンして、上下方向にドラッグします。
- 各視点を別の視点に切り替えたい時
視点の左上にあるアイコンをクリックして、切り替えたい視点のアイコンを選びます。(このチュートリアルガイドの作業中は、説明の中で指定された場合を除いて、切り替えないようにしてください。)
- 視点が「バードビュー」の場合、視点を回転させる時
キーボードの [Alt] を押しながら、「バードビュー」の上でマウスダウンして、視点を回転させたい方向にドラッグします。



バードビュー

ワークビューウィンドウでの作業中に、作業に都合のよいアングルになるように、これらの操作を行っててください。

ただし、視点が3次元ワールドに配置された「カメラ」キャストからの視点の場合、スライド等の操作はできませんので注意してください。

それでは、再度スコアウィンドウに説明を戻して、「ドア枠」、「ドア」、「ドアノブ」の3つのトラックに親子関係の設定を行いましょう。まず、「ドア」トラックを、「ドア枠」トラックの子として設定してみます。

親子リンクを設定する

■トラック間に親子リンクを設定する

次の手順を行ってください。

- 1) スコアウィンドウの「ドア」が登録されたトラックを、キーボードの「Shift」を押しながらマウスダウンします。
- 2) そのまま、「ドア枠」が登録されたトラックにドラッグ&ドロップします。



今の作業で、「ドア」モデルトラックが、「ドア枠」モデルトラックの子としてリンクされました。トラック名を見ると、親子関係を示すラインが表示されています。

2つのモデルトラックの親子関係の設定が終了しました。ワークビューウィンドウで見ると、2つのモデルキャストの中心位置が、3次元ワールドの中心に集まっていると思います。それぞれのモデルを正しい位置に配置しておきましょう。

■ 3次元ワールドのモデルを移動する

次の手順を行ってください。

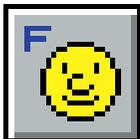
- 1) スコアウィンドウで、「ドア」トラックの0フレーム目をクリックして選択します。



- 2) ワークビューウィンドウで、上部にあるコマンドボタンの中から、「左右移動」をクリックして、押した状態にします。
- 3) 「正面」から表示された視点上でマウスダウンします。そのまま、カーソルを左にドラッグして、「ドア全体」モデルのドア枠と重なる位置に移動します。



左右移動



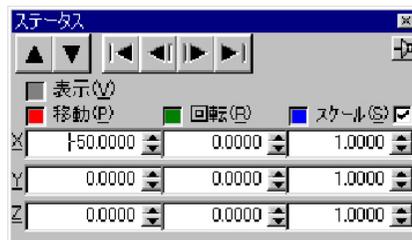
正面の視点



パース切り替え

上記の作業を行っても、正確に位置を合わせるのは難しい場合があります。

その際にはパース切り替えのボタンを押してパースの有無を調整してください。また、このような作業の場合、「ステータスウィンドウ」を使用すると正確に操作できます。



ステータスウィンドウ

「ステータスウィンドウ」は、3次元空間に配置されたキャストの中で、選択状態にあるトラックの、現在のフレームの「座標値」、「回転値」、「スケール値」を数値で表示するウィンドウです。数値を直接書き換える事により、値を変更する事ができます。

今までの作業を手順通りに進めて頂ければ、この「ドア」モデルの0フレーム目は、座標値 (Position) X = 「-50」になります。



モデルの配置作業について

モデリングビューウィンドウでパーツ分けしたした各モデルキャストを、リンク後にワークビューウィンドウを使って組み立てる場合には、どの位置に配置して良いのか分からなくなってしまう場合があります。

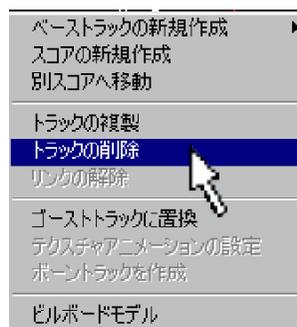
その際には、今行った作業のように、パーツ分けする前のモデルキャストを残しておいて、リンク後に組み立てる際に表示し、位置を参考にして頂くと正確に配置することができます。

ドアを構成するモデルキャストも全て配置が整いましたので、下絵として使用した「ドア全部」モデルの登録されたトラックを削除しておきましょう。

■トラックを削除する

次の手順を行ってください。

- 1) スコアウィンドウの、「ドア全部」モデルの登録されたトラック名をマウスでクリックして選択します。
- 2) 選択した「ドア全部」モデルのトラックでマウスの右ボタンを押します。表示されたプルダウンメニューから「トラックの削除」を選びます。



この作業は、スコアウィンドウから、選択したトラックを削除する作業ですので、「ドア全部」モデルキャスト自身が、プロジェクトの中から削除された訳ではありません。

ドアのキャラクターを組み立てることができました。しかし、ワークビューウィンドウに表示されるグリッドを地面として考えてみると、ドアが少し埋まった状態になっています。そこで親である「ドア枠」モデルを、次の手順で上に移動しておきましょう。

■ 3次元ワールドのモデルを移動する

次の手順を行ってください。

- 1) ワークビューウィンドウ上で「Ctrl」を押しながら、表示されている「ドア枠」モデルをクリックするか、またはスコアウィンドウの「ドア枠」モデルの登録されたトラック名を直接クリックして、「ドア枠」モデルのトラックを選択した状態にします。
- 2) ワークビューウィンドウで、上部にあるコマンドボタンの中から、「上下移動」をクリックして、押した状態にします。
- 3) 「正面」から表示された視点上でマウスダウンします。そのまま、カーソルを上ドラッグして、「ドア枠」モデルの底面とグリッドが重なる所まで、「ドア枠」モデルを移動します。



上下移動



正面の視点

今までの作業を手順通りに進めて頂ければ、この「ドア枠」モデルの0フレーム目は、座標値 (Position) Y = 「50」になります。今の作業のように、親となる「ドア枠」モデルが移動や回転などを行った場合、その子としてリンクされている「ドア」モデルは、親のモデルと同じように移動します。

お疲れさまでした！ドアモデルを組み立てる作業が全て終了しました。さっそく3次元空間に、「カメラ」と「ライト」のキャストを配置して、このドアモデルを再生画面で見てください。

カメラキャストを配置する

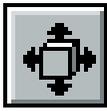
「カメラ」キャストは、あらかじめプロジェクトの中に1つ置かれています。

はじめにこの「カメラ」キャストを、「ドア」モデルを手前から撮影する位置へ移動してみましょう。先程の組み立て作業で行った手順と同じ操作です。

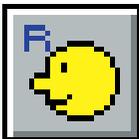
■ 3次元ワールドのカメラを移動する

次の手順を行ってください。

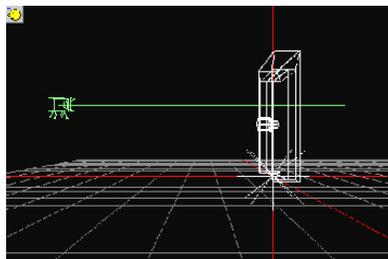
- 1) ワークビューウィンドウ上で、「Ctrl」を押しながら表示されている「カメラ0」キャストをクリックするか、またはスコアウィンドウの、「カメラ0」キャストの登録されたトラック名を直接クリックして、「カメラ0」キャストのトラックを選択した状態にします。
- 2) ワークビューウィンドウで、上部にあるコマンドボタンの中から「上下/左右移動」をクリックして、押した状態にします。
- 3) 「右面」から表示された視点上でマウスダウンします。そのままカーソルを左上へドラッグして、下の図のあたりにカメラを移動してください。



上下/左右移動



右の視点



参考までに、上の図の「カメラ0」キャストの0フレーム目は、座標値 (Position) X = 「0」、Y = 「100」、Z = 「-300」です。

次に、デフォルトで作成されているこの「カメラ0」キャスト自身の設定を変更しておきます。

■カメラのプロパティウィンドウを表示する

次の手順を行ってください。

- 1) キャストウィンドウのリストを、カメラ表示に切り替えるために、キャストウィンドウ上部にある「camera」と書かれたタブをマウスでクリックします。



- 2) カメラキャストリストの中の0番目に登録されている、「カメラ0」のキャスト名をマウスでダブルクリックします。

カメラのプロパティウィンドウが表示されました。

今回は、「カメラの焦点距離」設定を変更しておきます。

■カメラキャストの設定を変更する

次の手順を行ってください。

- 1) カメラプロパティウィンドウの、「カメラの焦点距離」設定を「50mm」から「15mm」に変更します。



- 2) カメラプロパティウィンドウを閉じます。

今回は、近距離からでも、ドアモデル全体を表示できるように、「カメラ0」キャストの表示する視野を広く設定しました。

ライトキャストを配置する

続いて、ライトキャストを用意しましょう。「環境光」キャストも、あらかじめプロジェクトの中に1つ置かれています。ライトは通常、「環境光」の他にも、「平行光」または「点光源」を1つ以上作っておくのが適切です。

今回は「平行光」を用意しておきましょう。

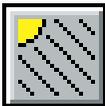
■ライトキャストを作成する

次の手順を行ってください。

- 1) ツールボックスの「プリミティブ」タブを選択し、ライト呼び出しボタンをマウスでクリックします。
- 2) 作成する光源の種類を指定する、「ライトの新規作成」ダイアログボックスが表示されます。
今回は「平行光」をクリックします。



ライト呼び出し



平行光

「ライト」キャストが作成されました。プロパティウィンドウから呼び出されるキャストは、作成された時に、スコアウィンドウの新規トラックに登録されます。また、同時にキーフレームも自動的に作成されます。

(先程、立方体から制作した「ドア全部」モデルが、スコアウィンドウに登録されていたのもそのためです。)

「平行光」は、3次元ワールド全体を一定の方向から照らす光です。そのため、配置する場所を変えても意味がありません。方向だけ変更しておきましょう。

■3次元ワールドのライトを回転する

次の手順を行ってください。

- 1) ワークビューウィンドウ上で「Ctrl」を押しながら「平行光 0」をクリックするか、またはスコアウィンドウの「平行光 0」の登録されたトラックを直接クリックして、「平行光 0」キャストのトラックを選択した状態にします。



Z軸回転

- 2) ワークビューウィンドウで、上部にあるコマンドボタンの中から、「Z軸回転」をクリックして、押した状態にします。
- 3) いずれかの視点上でマウスダウンします。そのまま平行光を回転させたい方向へドラッグします。(1つの視点でうまく回転できない場合は、様々な視点から回転を行ってください。)



「平行光0」キャストを回転させて、上の図のような角度から光をあてました。この場合、3次元ワールド全体のモデルを、斜め上から照らします。

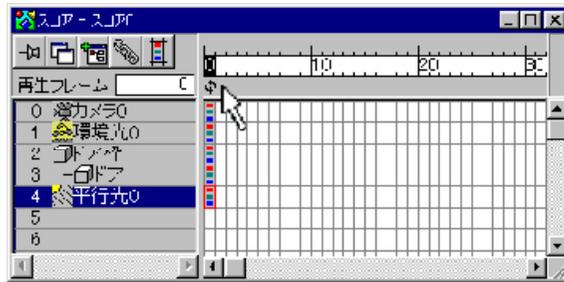
参考までに、上の図の「平行光0」キャストの0フレーム目の値は、回転値 (Rotate) X = 「45」、Y = 「-45」、Z = 「0」です。

カメラとライトの準備もできました。後はこのフレームを再生するだけです。再生する場合は、基本的に0フレーム目から順番に再生していきます。この状態では、0フレーム目を描画するのが最初の一瞬だけになってしまいます。そこで0フレーム目だけを再生するようにするために、スコアに「ループ枠」を設定しておきましょう。

■ループ枠を設定する

次の手順を行ってください。

- 1) スコアウィンドウの、ループ枠を設定するトラックの0フレーム目をダブルクリックします。(次の図を参考にしてください。)



これで、0フレーム目にループ枠を設定することができました。

プロジェクトを再生する準備が整いましたので、ツールボックスの「制御パネルウィンドウ」タブの「再生」ボタンを押してみましょう。

プロジェクトの再生、保存

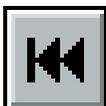


ツールボックス「制御パネル」

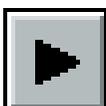
■プロジェクトを再生する

次の手順を行ってください。

- 1) 再生画面を「ステージビューウィンドウ」に設定するために、制御パネルウィンドウのモードを、「ステージビュー再生モード」に切り替えます。(下の図の中心にあるボタンが押されていることを確認してください。)



先頭へ移動



再生



- 2) 「先頭へ移動」ボタンを押して再生フレームを0フレーム目に巻き戻してから、「再生」ボタンを押します。

再生を表示する「ステージビューウィンドウ」が開かれ、プロジェクトが再生されます。

ステージビューウィンドウを閉じてください。
ここで、一度このプロジェクトを保存しておきましょう。

■プロジェクトを上書き保存する

次の手順を行ってください。

- 1) メニューの「プロジェクト」から「プロジェクトの上書き保存」を選びます。

このプロジェクトファイルは、次からの章での説明にも使用しますので、削除しないでください。

ドアモデルの制作を通して、プロジェクトの中に、モデルキャストやテクスチャキャスト（ビットマップも同様）等の素材を用意する方法を説明してまいりました。

モデルキャストを組み立てる方法も、お分かり頂けたかと思います。

ここまでの作業は、素材を用意する地道な作業が多かったのですが、次の章では、このドアモデルに3次元アニメーションを設定する作業を行っていきます。

第2章 アニメーションの作成

2章では

第2章では、プロジェクトの中に用意したキャストに、アニメーションを設定する作業を中心に学習します。

■この章で学習する内容

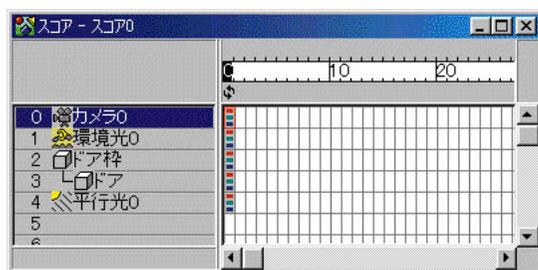
- モデルキャストにアニメーションを設定する
- アニメーションに効果音を付ける
- 複数のアニメーションを同時に再生する

モデルキャストにアニメーションを設定する

それでは、1章で作成した「Door.lcp」のプロジェクトファイルを開いてください。このプロジェクトの中に作成された、ドアモデルに、アニメーションを設定していきましょう。

複数のキーフレームを作成する

まず、スコアウィンドウを見てください。スコアウィンドウのトラックに、「ドア枠」、「ドア」、「ドアノブ」モデルキャスト、「カメラ0」キャスト、2種類のライトキャストが登録されています。



それぞれのトラックの0フレーム目には、3次元ワールドでの位置、方向、サイズ、可視を決定するための「キーフレーム」が作成されています。



この、スコアウィンドウの「キーフレーム」の表示について、少し詳しく説明します。



赤 = Position (座標値) のキーフレーム
緑 = Rotate (回転値) のキーフレーム
青 = Scale (スケール値) のキーフレーム

座標値、回転値、スケール値のキーフレームが設定されているフレームは、座標値のキーフレーム = 「赤」、回転値のキーフレーム = 「緑」、スケール値のキーフレーム = 「青」のマークが表示されます。

また、可視 (見えるようにする) のキーフレームが設定されているフレームは、フレームが灰色で表示されます。

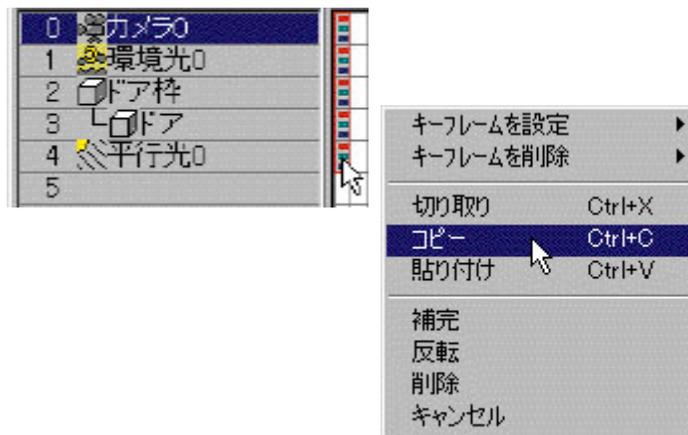
3次元ワールド内でアニメーションを設定するには、このキーフレームを設定する作業が重要となってきます。

現在、0フレームのみにキーフレームが設定されていますので、先のフレームにもキーフレームを設定していきましょう。

■キーフレームを複製する

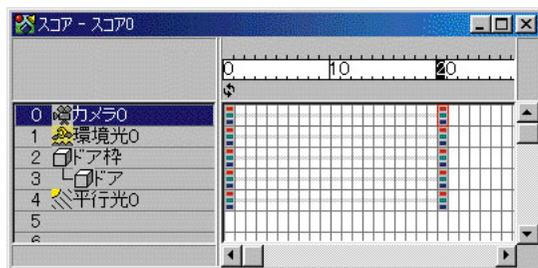
次の手順を行ってください。

- 1) スコアウィンドウで、キャストが登録されたトラックのキーフレームを、マウスダウンしながら、カーソルを上から下にドラッグします。(下の図を参考にしてください。)



- 2) 一番下のキーフレームまできたら、マウスの左ボタンを離します。カーソルが手の形であることを確認して、マウスの右ボタンをクリックし、表示されるプルダウンメニューから「コピー」を実行します。

- 3) 赤い枠で囲われたフレーム上にカーソルを持っていくと、カーソルが手の形の表示に変わります。その状態でマウスダウンして、コピーしたフレームを、20フレームまで右へドラッグして移動します。
- 4) 20フレーム目まで移動したら、赤い枠に囲まれたフレーム以外のフレーム上を一度クリックし決定します。

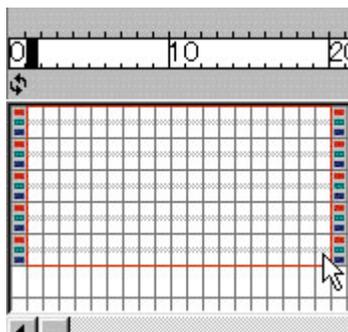


0フレーム目に作成した全キーフレームを、20フレーム目に複製しました。次は、この0フレーム目から20フレーム目までの「可視」のキーフレームを、全て設定します。

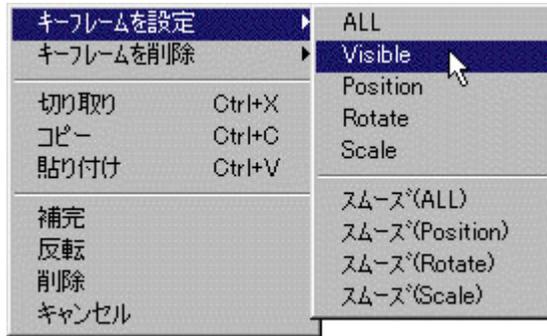
■可視のキーフレームを設定する

次の手順を行ってください。

- 1) スコアウィンドウで、「カメラ0」キャストトラックの1フレーム目でマウスダウンします。そのまま、下の図を参考に、カーソルをドラッグします。「平行光0」キャストトラックの19フレーム目までをドラッグして、赤い枠で囲みます。



- 2) 囲んだら、マウスの左ボタンを離します。カーソルが手の形であることを確認して、マウスの右ボタンをクリックし、表示されるプルダウンメニューから、「キーフレームを設定／表示」を実行します。



- 3) 赤い枠に囲まれたフレーム以外の、フレーム上を一度クリックし決定します。

選択した全フレームの「可視」のキーフレームが設定され、灰色で表示されました。

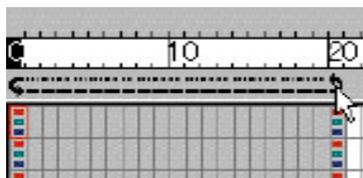
ここまでの作業で、トラックに登録されたキャストが、20フレーム目まで表示されるようになりました。

次に、0フレーム目をループするように設定してある「ループ枠」を、20フレーム目までループ再生するように、引き伸ばします。

■ループ枠を引き伸ばす

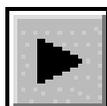
次の手順を行ってください。

- 1) スコアウィンドウの、ループ枠を設定するトラックの、0フレーム目に作成されたループ枠をマウスでダブルクリックします。そのまま右へドラッグして、ループ枠を20フレームまで引き伸ばします。(下の図を参考にしてください。)





先頭へ移動



再生

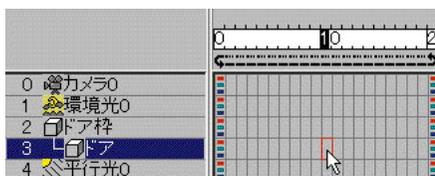
ここで、一度プロジェクトを再生してみましょう。制御パネルウィンドウの「先頭へ移動」ボタンを押してから、「再生」ボタンを押してください。

この段階では、再生画面に動きの変化は見られません。それでは「ドア」モデルの途中フレームに、新たなキーフレームを作成してみましょう。

■回転のキーフレームを設定する

次の手順を行ってください。

- 1) スコアウィンドウで、「ドア」モデルキャストの登録されたトラックの10フレーム目をマウスでクリックします。



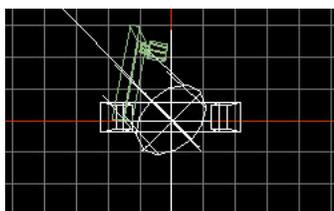
- 2) ワークビューウィンドウで、上部にあるコマンドボタンの中から、「Z軸回転」をクリックして、押した状態にします。
- 3) 「上面」から表示された視点上でマウスダウンします。そのまま、カーソルを下方方向にドラッグして、「ドア」モデルの10フレーム目を回転値 (Rotate) Y = 「-80」になる所まで回転します。



Z軸回転



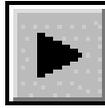
上面の視点



10フレーム目の「ドア」モデルの回転値を変更しました。スコアウィンドウの、「ドア」モデルトラックの10フレーム目を見ると、自動的に「回転値」のキーフレームが設定されています。



先頭へ移動



再生

それでは、再度プロジェクトを再生してみましょう。
制御パネルウィンドウの「巻き戻し」ボタンを押してから、「再生」ボタンを押してください。

今度はドアが開くアニメーションを行いました。ここで、この回転アニメーションが作られた理由を説明します。

先程「ドア」モデルの0フレーム目と20フレーム目に、ドアが閉じた状態のキーフレームを設定しました。次に途中の10フレーム目に、ドアが開いた状態の回転のキーフレームを設定しました。DigitalLocaでは、キーフレームとキーフレームの間の数値を自動的に中割りします。このことにより次第にドアが開閉するアニメーションが作られたのです。

この中割りアニメーションは、座標やスケールのキーフレームでも同様に操作できますが、今回は回転のキーフレームで説明しました。



可視のキーフレームについて

モデル、カメラ、ライト等のキャストが、フレームごとに表示するかどうかを指定する、「可視」のキーフレームについて、もう少し詳しく説明します。

可視のキーフレームは、そのキャストトラックの可視不可視を決定しますが、同時にそのトラックの子としてリンクされているトラックの可視不可視も決定します。入門で制作したドアのモデルで説明すれば、親としてリンクしている「ドア枠」トラックの、「可視」キーフレームが不可視の設定であれば、子となる「ドア」モデルトラックも不可視の状態になります。

アニメーションに効果音を付ける

先程作成したドアの開くアニメーションに、効果音をつけてみましょう。まず、効果音のキャストとなる「Wave サウンドキャスト」をプロジェクトの中に作成します。素材となる「音」のデータは、外部ファイルから読み込みます。

■ Wave サウンドキャストを作成する

次の手順を行ってください。

- 1) キャストウィンドウのリストを、Wave サウンドキャスト表示に切り替えるために、キャストウィンドウ上部にある「wave」と書かれたタブをマウスでクリックします。

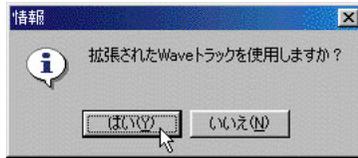


- 2) Wave サウンドキャストリストの中のもの、何も登録されていない欄をマウスでダブルクリックします。
- 3) 「ファイルの読み込み」ダイアログボックスが表示されますので、DigitalLocaがインストールされたディレクトリの「Tutorial¥Walker¥Wave¥Open.wav」を選択して読み込みます。

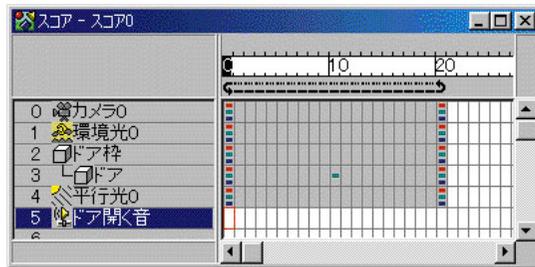
「...を外部ファイルにしますか?」というダイアログボックスが表示されますので、「いいえ」を選んでください。

「Wave」フォーマットのサウンドデータを読み込んで、「Wave」サウンドキャストが作成されました。この「Open」キャストの名前は「ドア開く音」という名前に変更しておいてください。

続いて、スコアウィンドウの新規トラックにドラッグ&ドロップして登録してください。



「拡張された Waveトラックを使用しますか」というダイアログボックスが表示されますので、「はい」を選んで下さい。ここで、サウンドの編集をするかどうかを決定します。



トラックに登録された「Wave」サウンドキャストの、音を再生するフレームを指定します。

■ Wave サウンドテキストの再生フレームを指定する

次の手順を行ってください。

- 1) スコアウィンドウの、「ドア開く音」サウンドキャストが登録されたトラック上で、ドアが開き始める、1フレーム目をマウスでダブルクリックします。(下の図を参考にしてください。)



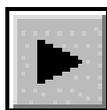
「再生制御」ウィンドウが表示されました。このウィンドウで音の効果を編集していきます。



今回は音を一度再生するだけですので、「再生制御」をチェックし「Play」に設定されている事を確認してください。設定が終わったら「OK」を押して決定してください。このウィンドウの効果を上手く利用する事で様々なサウンド効果が得られます。



先頭へ移動



再生

それでは、プロジェクトを再生してください。制御パネルウィンドウの「巻き戻し」ボタンを押してから、「再生」ボタンを押してください。

ドアが開くアニメーションと同時に、ドアの開く音が再生されるようになりました。

複数のアニメーションを同時に再生する

次に、ドアモデルを複製し、それらに別々のアニメーションを設定する方法を用いて、DigitalLocaの「マルチスコア」機能の概念を説明していきます。

まずはじめに、1つのプロジェクトに複数のドアモデルを用意するため、ドアを構成するモデルトラックを複製しておきましょう。

■トラックを複製する

次の手順を行ってください。

- 1) スコアウィンドウの、「ドア枠」モデルの登録されたトラックをマウスでクリックして選択します。
- 2) 選択した「ドア枠」のトラック名でマウスの右ボタンを押します。表示されるプルダウンメニューの中から「トラックの複製」を選びます。

「トラックの複製」ダイアログボックスが表示されますので、複製する数値を入力してください。

今回は2つのドアを使用しますので、トラック数は1と入力し「OK」を押して下さい。



スコアウィンドウの新規トラックに、ドアを構成するモデルトラックが全て複製されました。

続いて、複製されたドアモデルのアニメーションを少し長くしてみましよう。

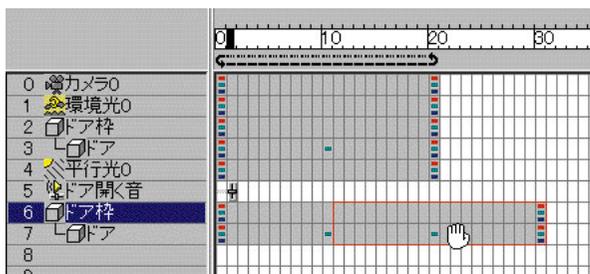
■キーフレームを複製する

次の手順を行ってください。

- 1) スコアウィンドウで、下の図を参考に、キーボードの [Ctrl] を押しながら、複製された「ドア枠」トラックの1フレーム目でマウスダウンします。そのまま「ドア」トラックの、20フレーム目までを囲うようにカーソルをドラッグします。



- 2) キーボードの [Ctrl] とマウスの左ボタンを離します。カーソルが手の形である事を確認し、マウスの左ボタンを押します。下の図を参考に、後ろのキーフレームが30フレーム目に来る所まで、右へドラッグして移動します。

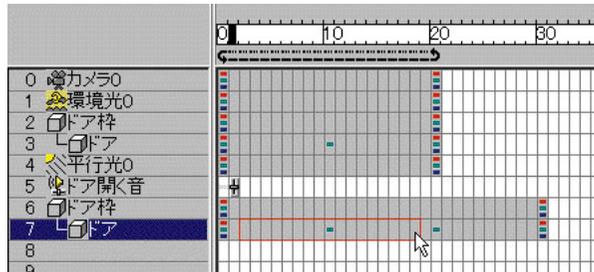


- 3) 上の図の所へ移動したら、赤い枠に囲まれたフレーム以外の、フレーム上を一度クリックし決定します。

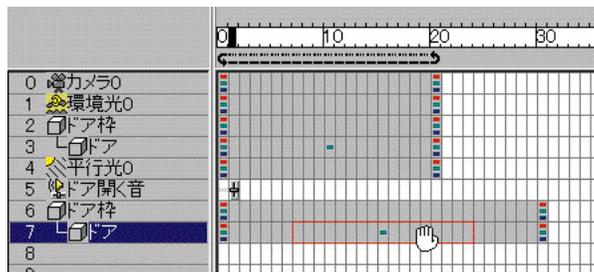
キーボードの「Ctrl」を押しながらフレームを囲う方法は、コピーをするためのショートカットです。

第2章 - アニメーションの作成

- 4) 再度、下の図を参考に、キーボードの [Ctrl] を押しながら、複製された「ドア」トラックの2フレーム目でマウスダウンします。そのまま「ドア」トラックの18フレーム目までを囲うように、右へカーソルをドラッグします。



- 5) キーボードの [Ctrl] とマウスの左ボタンを離します。カーソルが手の形であることを確認し、マウスの左ボタンを押します。下の図を参考に、回転のキーフレームが15フレーム目に来る所まで、カーソルを右へドラッグして移動します。



- 6) キーフレームが図の所へ移動したら、赤い枠に囲まれたフレーム以外の、フレーム上を一度クリックし決定します。

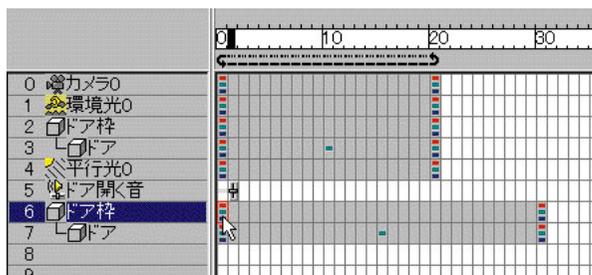
フレーム上のキーフレームを複製移動して、ドアが開くアニメーションを、10フレーム長くしました。

次は、この複製されたドアモデルが、元のドアモデルと同じ座標にありますので、少し移動しておきます。

■ 3次元ワールドのモデルを移動する

次の手順を行ってください。

- 1) スコアウィンドウで、複製したドアモデルの「ドア枠」トラックの、0フレーム目のキーフレームをクリックして選択します。



- 2) ステータスウィンドウで、選択されている0フレーム目の座標値を、座標値 (Position) X = 「150」に書き換え、キーボードの「Enter」を押します。



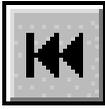
複製したドアモデルの「ドア枠」トラックの0フレーム目を右へ移動しました。

同じように30フレーム目のキーフレームも、同じ座標にしておきましょう。

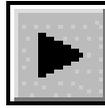
- 3) 「ドア枠」トラックの30フレーム目をクリックして選択します。



- 4) 同じく、ステータスウィンドウで、選択されている30フレーム目の座標値を、座標値 (Position) X = 「150」に書き換え、キーボードの「Enter」を押します。



先頭へ移動



再生

複製したドアキャラクターを横へ移動しました。

ここで、一度プロジェクトを再生してみてください。制御パネルウィンドウの「先頭へ移動」ボタンを押してから、「再生」ボタンを押してください。

左側のドアモデルのアニメーションは問題ありませんが、複製したドアモデルのアニメーションは、最後まで再生されず途中で0フレームに戻ってしまいます。

これは、アニメーションの再生フレーム数の異なるキャラクターが、1つのスコアウィンドウに登録されているためです。

この場合、ループ枠が、0～20フレームの間に設定されていますので、それ以上長いアニメーションは、途中までしか再生されません。

そこで、別のスコアウィンドウを作成して、片側のドアモデルを別スコアへ移動させておきましょう。今回は、複製したドアモデルを、新規スコアへ移動します。

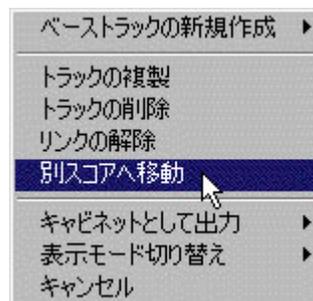


マルチスコア機能を使用する

■新規スコアウィンドウを作成する

次の手順を行ってください。

- 1) スコアウィンドウの、複製したドアモデルの親トラックである「ドア枠」モデルのトラック名をマウスでクリックして選択します。
- 2) 選択した「ドア枠」モデルのトラック名でマウスの右ボタンを押します。表示されるプルダウンメニューから「別スコアへ移動」を選択します。



- 3) 「別のスコアへ移動」ダイアログボックスが表示されますので、「(新規スコア)」をクリックして選択します。最後に「OK」ボタンをクリックします。

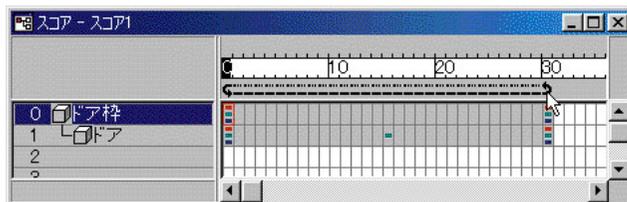
プロジェクト内に、新しいスコアが追加され、その中に、「ドア枠」トラックとその子としてリンクしている「ドア」トラックが移動しました。新しく作られたスコアウィンドウには、「スコア 1」という名前が付けられています。

それでは、今移動したドアモデルに、専用のループ枠を設定しておきましょう。

■ループ枠を設定する

次の手順を行ってください。

- 1) 追加された「スコア 1」スコアウィンドウの、ループ枠を設定するトラックの0フレーム目をダブルクリックします。
- 2) 作成されたループ枠でマウスダウンし、そのまま左へドラッグして30フレーム目まで引き伸ばします。



ここで、再度プロジェクトを再生してみてください。制御パネルウィンドウの「先頭へ移動」ボタンを押してから、「再生」ボタンを押してください。



先頭へ移動



再生

今度は、それぞれのキャラクターが、自分の登録されたスコアウィンドウのループ枠にそってフレームを再生していますので、正しくアニメーションを繰り返しています。

DigitalLoca では、このように、1つのプロジェクトの中に複数のスコアウィンドウを持たせることができます。

このマルチスコア機能は、次の章で説明するスクリプト機能と合わせて使用することによって、さらに実力が発揮されます。

複製したドアモデルの登録された、「スコア 1」スコアウィンドウの新規トラックにも、Wave サウンドキャストの「ドア開く音」キャストを登録すれば、こちらのドアモデルにもドアを開く効果音を付けることができます。

今回このチュートリアルガイドの中では行いませんが、先程の説明を参考にして、ぜひ両方のドアモデルが、ドアが開く効果音を再生するようにしてみてください。

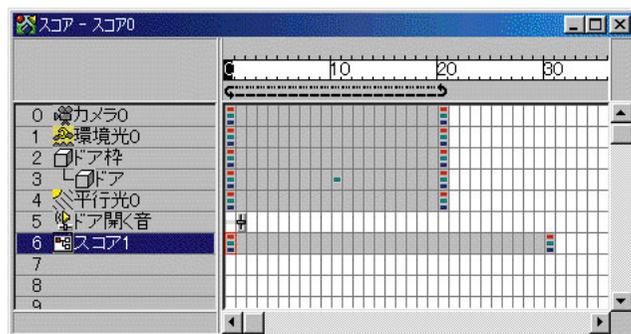
作業が終了したら、一度プロジェクトを保存しておいてください。また、このプロジェクトファイルは次からの章でも説明に使用しますので、削除しないでください。

マルチスコア機能について

複数のスコアを同時に管理する事のできる「マルチスコア機能」について、補足の説明をします。次の章へ進む前に必ず読んでおいてください。

複数のスコアをプロジェクトの中に作成しても、はじめからプロジェクトの中に用意されているスコアウィンドウが、あくまで全ての「親」のスコアウィンドウとなります。

スコアウィンドウを追加作成すると、「親スコア」の中に、追加されたスコアの「親トラック」が作成されます。



「可視」のキーフレームを例にあげて説明しますと、「親スコアウィンドウ」に登録された「親トラック」の、不可視に設定されているフレームでは、その子となるスコア内のフレームも不可視となります。同じように、追加作成されたスコアウィンドウの中に、モデルやカメラ、ライトを登録して「可視」のキー

フレームを設定しておいても、「親スコアウィンドウ」に登録されているスコアの「親トラック」の可視が設定されていなければ、可視の状態になりません。

また、「親スコアウィンドウ」に作成される「親トラック」自身も、3次元ワールドでの座標値、回転値、スケール値を持っています。

先程の作業のように、新規スコアを作成すると同時に、モデルトラックを別スコアへ移動させる場合では、移動させるモデルトラックに登録された最後のキーフレームまで、自動的に「親スコアウィンドウ」の「親トラック」に「可視」のキーフレームが設定されます。

第3章 スクリプトの作成

3章では

第3章では、プロジェクトにスクリプトと呼ばれるプログラムを作成して、作品にインタラクティブ性をもたせる作業を中心に学習します。

■この章で学習する内容

- キャストクリックイベントを使用する
- キャビネットファイルを使用する
- 当り判定と条件分岐命令を使用する

キャストクリックイベントを使用する

スクリプトキャストは、プロジェクトの再生を制御するプログラムです。このプログラムを使用すると、プロジェクトを再生中に、こちら側からの操作によって変化するアニメーションを作ることができます。

プログラムを作った経験がない人でも、簡単な命令文を少し覚えるだけで、十分に楽しい作品が制作できます。

マウスクリックに反応させる

はじめに、再生画面に表示されるキャラクターをマウスでクリックすると、アニメーションが変化するという作品を作ってみます。まず、2章で作成した「Door.lcp」のプロジェクトファイルを開いてください。

スクリプトキャストの説明に入る前に、先程「マルチスコア機能」を説明するために作成した、「スコア 1」スコアウィンドウを削除しておきましょう。もし、現在のプロジェクトの状態を保存しておきたい場合は、ここで「Door.lcp」ファイルを複製しておくか、今のプロジェクトを別名で保存しておいてください。

それでは、「スコア 1」スコアウィンドウを削除する作業を行います。

■サブスコアを削除する

次の手順を行ってください。

- 1) メニューの「プロジェクト」から「スコアの削除」を選びます。
- 2) 「スコアの削除」ダイアログボックスが表示されますので、「スコア 1」をマウスでクリックして「OK」を押します。

別スコアを作成する前の状態に戻りました。

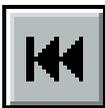
続いてスコアウィンドウを見てください。0～20フレーム目までをループさせるループ枠が設定されています。ここで、再生フレームを、ドアが閉まっている状態の0フレーム目だけでループ再生するように変更しておきましょう。

■ループ枠を削除する

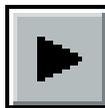
次の手順を行ってください。

- 1) スコアウィンドウの、0～20 フレーム目までに登録されているループ枠をダブルクリックして削除します。
- 2) ループ枠を設定するトラックの0 フレーム目をダブルクリックします。0 フレーム目にループ枠が設定されました。

上記の作業の代わりに、0～20 フレーム目までに登録されているループ枠の一番右端でマウスダウンし、そのまま0 フレーム目までドラッグして、0 フレーム目のみでループするように設定するという方法もあります。



先頭へ移動



再生

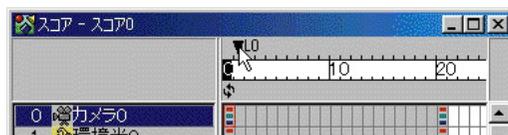
それではここで、プロジェクトを再生してみましょう。制御パネルウィンドウの「先頭へ移動」ボタンを押してから、「再生」ボタンを押してください。

0 フレーム目にループ枠が設定されていますので、先のフレームにあるドアが開くアニメーションを再生しなくなりました。続いて、スコアのフレームに名前を付けるための「ラベル」を作成します。

■ラベルを設定する

次の手順を行ってください。

- 1) スコアウィンドウの、ラベルを設定するトラックの1 フレーム目を、マウスでダブルクリックします。ダブルクリックしたフレーム上にラベルが作成されました。



- 2) 続いて、作成された三角形のラベルマークをマウスでクリックします。「L0」とある名前を、「ドア開く」に書き換えて、キーボードの「Enter」を押します。

1 フレーム目に、「ドア開く」という名前のラベルを設定しました。このラベルは、スクリプトプログラムで、フレームの指定をする際に役立ちます。

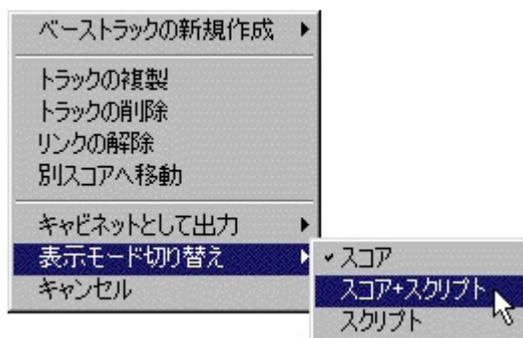
続いて、モデルトラックのフレーム上にスクリプトキャストを設定してみましょう。

はじめに、スコアウィンドウの表示モードを切り替えます。

■スコアウィンドウの表示モードを切り替える

次の手順を行ってください。

- 1) スコアウィンドウの、トラック名が表示されている場所（どの欄でも良い）にカーソルを移動して、マウスの右ボタンを押します。
- 2) 表示されるプルダウンメニューから「表示モード切り替え／スコア+スクリプト」を選びます。



スコアウィンドウのフレーム表示が、スコアとスクリプトの両方の設定を表示するモードに切り替わりました。

ー DigitalLoca 3.0 では、スコア上部にある「表示モードの切り換え」ボタンでも切り替えることができます。

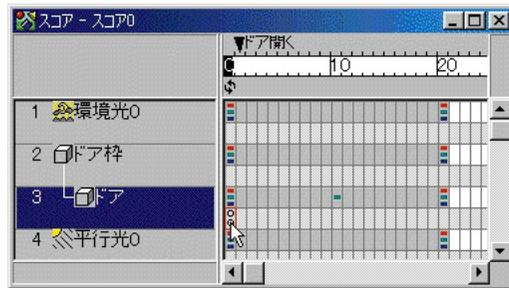
スコアウィンドウを、「ドア」モデルトラックが見える状態にしてください。

「ドア」モデルトラックの0フレーム目に、新規スクリプトキャストを設定してみましょう。

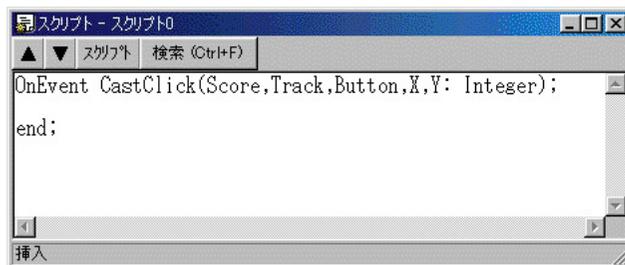
■トラックにスクリプトを登録する

次の手順を行ってください。

- 1) スコアウィンドウの、「ドア」モデルトラックの、0フレーム目の下段をマウスでダブルクリックします。



- 2) ダブルクリックしたフレームに、スクリプトキャストが設定されます。同時に、設定されたスクリプトキャストのプロパティウィンドウが表示されます。



スクリプトプロパティウィンドウ

図のウィンドウが、スクリプトキャストのプログラムを入力するプロパティウィンドウです。

すでにプログラムが書かれていますが、その説明は後半のスクリプト入門の項で行います。

このウィンドウの中にテキスト入力をし、プログラムの作成をしていきましょう。

■トラックにスクリプトを登録する

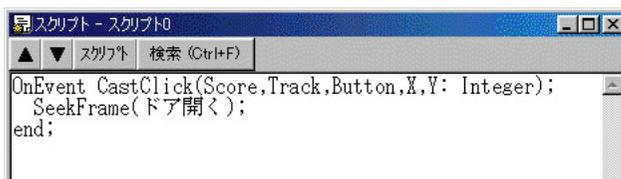
次の手順を行ってください。

- 1) 先ほどの作業で、フレーム上に設定されているスクリプトキャストのプロパティウィンドウが開かれています。このウィンドウの1行目と2行目の間の行に、次のプログラムを入力します。

SeekFrame(ドア開く);

- 2) プログラムを入力する際の注意点として、「ドア開く」以外のテキストは、全て半角英数文字で入力してください。

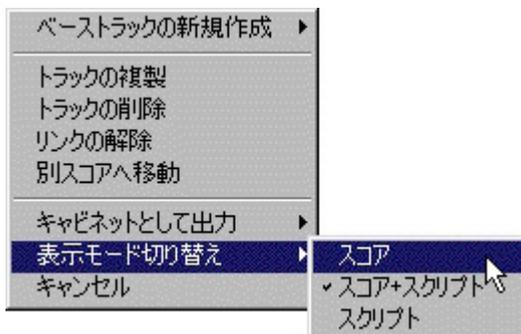
1行目と2行目の間に、上のテキストを入力して、スクリプトキャストを下の図のように書き換えました。



このプログラム命令の意味を説明します。

『このスクリプトを設定したフレームを再生中に、もしこのキャストがマウスクリックされたら、再生フレームを「ドア開く」ラベルの付いたフレームへジャンプさせなさい。』という内容です。

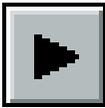
「ドア」モデルトラックにスクリプトを設定できましたので、スクリプトウィンドウを閉じて、再度スコアウィンドウの表示モードをスコア表示の状態に戻しておいてください。



最後に一時的な処置として、再生フレームが、アニメーション設定のされていない先のフレームへ進まないようにするために、スコアウィンドウの20フレーム目にもループ枠を設定しておいてください。



初期化



再生

それではプロジェクトを再生してみましょう。今回からは、制御パネルウィンドウの「先頭へ移動」ボタンの代わりに、「初期化」ボタンを押してから、「再生」ボタンを押してください。「初期化」ボタンを押すことによりスクリプトの構文チェックを自動的に行います。

相変わらずドアは閉まったままです。そこで、再生画面のドアの部分をマウスでクリックしてみてください。すると、ドアが開くアニメーションを行いました。これは、「ドア」モデルトラックの0フレーム目に設定したスクリプトプログラムが、モデルをクリックすることによって実行されたためです。再度、ドアをクリックしてみてください。今度はドアが開くアニメーションを行いません。これは、最後の20フレーム目にループ枠を設定しているためで、「ドア」モデルトラックの20フレーム目には、先程のスクリプトプログラムが設定されていないためです。

この状態では、再生してから一度しかドアを開くアニメーションを行いません。そこで、20フレーム目を編集して、新たなスクリプトを設定してみましょう。

まず、先程一時的に設定した、20フレーム目のループ枠をダブルクリックして削除します。

次に、スコアの0フレーム目に、新しいラベルを作成します。

■ラベルを作成する

次の手順を行ってください。

- 1) スコアウィンドウの、ラベルを登録するトラックの0フレーム目を、マウスでダブルクリックします。ダブルクリックしたフレーム上にラベルが作成されました。
- 2) 続いて、作成された三角形のラベルマークをマウスでクリッ



クします。「LO」とある名前を、「ドア静止」に書き換えて、キーボードの「Enter」を押します。

0フレーム目に、「ドア静止」という名前のラベルが設定されました。このラベルは、次に作成するスクリプトプログラム命令で使用します。

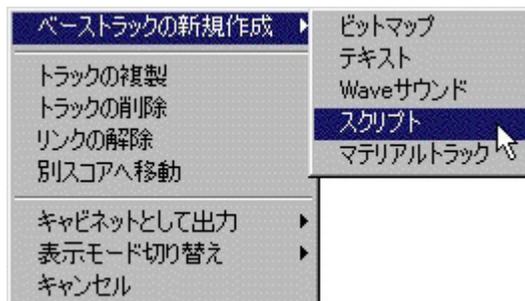
今回は、先程とは異なる方法でスクリプトを設定してみます。それではこのスコアウィンドウに、スクリプトトラックを用意してみましょう。

■スコアにスクリプトトラックを作成する

次の手順を行ってください。

- 1) スコアウィンドウの、トラック名が表示されている所にカーソルを移動して、マウスの右ボタンを押します。
- 2) 表示されるプルダウンメニューから「ベーストラックの新規作成／スクリプト」を選びます。

スコアウィンドウの新規トラックに、「スクリプトトラック」が



作成されました。スクリプトトラックに設定されたスクリプトプログラムも、そのフレームが再生される時に実行されます。スクリプトトラックの20フレーム目に、新しいスクリプトキャストを作成して設定してみましょう。

■スクリプトトラックにスクリプトを設定する

次の手順を行ってください。

- 1) スコアウィンドウの、スクリプトトラックの、20フレーム目をマウスでダブルクリックします。
- 2) ダブルクリックしたフレームに、スクリプトキャストが設定



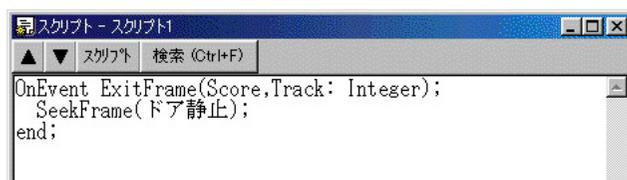
されます。同時に、設定されたスクリプトキャストのプロパティウィンドウが表示されます。

- 3) 開かれたスクリプトキャストのプロパティウィンドウの1行目と2行目の間に、次のプログラムを入力します。

Seekframe(ドア静止);

1行目と2行目の間に、上のテキストを入力して、スクリプトキャストを下の図のように書き換えました。

このプログラム命令の意味を説明します。

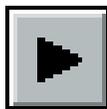


『このスクリプトが設定されたフレームを再生した後に、再生フレームを「ドア静止」ラベルの付いたフレームへジャンプさせなさい。』という内容です。

それでは、プロジェクトを再生してみましょう。制御パネルウィンドウの「初期化」ボタンを押してから、「再生」ボタンを押してください。



初期化



再生

先程と同様に、再生画面のドアの部分をマウスでクリックすると、ドアが開くアニメーションを行います。

再度、ドアをクリックしてみてください。今度は何度クリックしても、ドアが開くアニメーションを行います。これは、20フレーム目が再生された時に、スクリプトトラックの20フレーム目に設定したスクリプトが実行され、「ドア静止」ラベルの設定されたフレームへジャンプしているためです。

ここで、一度プロジェクトを保存しておいてください。

このプロジェクトファイルは次からの説明にも使用しますので、削除しないでおいてください。

補足として、スコアウィンドウ内で、作成し設定するスクリプトキャストも、キャストウィンドウのスクリプトキャストのリスト内に並んでいます。

キャビネットファイルを使用する

いよいよこのチュートリアルガイドも作品を完成させる部分の説明に入ってきました。

ドアモデルを題材にして、モデルの作成、アニメーションの作成、スクリプトの作成と、DigitalLoca のプロジェクトを作成していく上での、基本的な作業の流れをつかんで頂けたと思います。それでは最後にもう少し、DigitalLoca の機能やスクリプトの使用例を説明させていただきます。

別プロジェクトに移動する

はじめに、このプロジェクトの中のドアモデルトラックを、別のプロジェクトの中に取り込むために、プロジェクト同士がデータ交換用に使用する「キャビネットファイル」として書き出します。

■トラックをキャビネットファイルとして出力する

次の手順を行ってください。

- 1) スコアウィンドウの、ドアモデルの親トラックである「ドア 枠」モデルのトラック名をマウスでクリックして選択します。
- 2) 選択した「ドア 枠」のトラック名でマウスの右ボタンを押します。表示されるプルダウンメニューから「出力／キャビネットファイル」を選びます。



- 3) 「確認」ダイアログボックスが表示されますので、「いいえ」を選択します。



- 4) キャビネットファイルを保存する先を指定し、ファイル名を「Door」と入力して「保存」を押します。

「Door.lco」という名前のキャビネットファイルが保存されました。続いて、新たなプロジェクトを開いてみましょう。

■プロジェクトファイルを開く

次の手順を行ってください。

- 1) メニューの「プロジェクト」から「プロジェクトを開く」を選びます。
- 2) 「ファイルを開く」ダイアログボックスが表示されますので、DigitalLoca がインストールされたディレクトリの「Walker¥Project¥Walker.lcp」を選択して読み込みます。

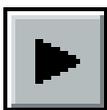
「Walker.lcp」というプロジェクトが開かれました。一度、このプロジェクトを再生してみてください。このプロジェクトは、キーボードのカーソルキーによって、緑色のキャラクター「ウォーカー」を操作することのできるプロジェクトです。

これは、スクリプトプログラムの「パペット機能」を使用したサンプルです。「パペット機能」のスクリプトに関しましては、後半のスクリプト入門の項で詳しく説明いたしますが、簡単に説明しますと、モデルトラックの座標値や回転値等を、スクリプトによって変化させるといったプログラムです。

これからこのプロジェクトの中に、あなたの作成したドアモデルと、ドアが開くアニメーションを取り込みます。



初期化



再生

先程作成したスクリプト命令では、再生画面のドアノブをクリックすることによって、ドアが開くアニメーションフレームにジャンプさせました。

今度はこのプロジェクトの中で、スクリプトの「当り判定」命令を使用して、ドアが開くアニメーションフレームへジャンプさせてみます。

それでは、先程作成したドアモデルをプロジェクトの中に読み込んでみましょう。

■キャビネットファイルを入力する

次の手順を行ってください。

- 1) メニューの「プロジェクト」から「入力／キャビネットファイル」を選びます。
- 2) 「ファイルを開く」ダイアログボックスが表示されますので、先ほど保存したキャビネットファイル「Door.lco」を選択して読み込みます。
- 3) 「読み込み先の選択」ダイアログボックスが表示されますので、「ルートスコア」を選択し「OK」を押します。



ドアモデルを構成するモデルキャスト、テクスチャキャスト、トラック情報が読み込まれました。

このプロジェクトには、現在5つのスコアウィンドウが存在していますが、はじめからプロジェクトに存在する親スコアウィンドウは、「ルートスコア」という名前が付けられたスコアウィンドウです。

親スコアであるルートスコアウィンドウを見てください。読み込んだドアモデルのトラックが作成されています。

このドアモデルにも専用のスコアウィンドウを作成しておきましょう。

■新規スコアウィンドウを作成する

次の手順を行ってください。

- 1) ドアモデルの、親トラックである「ドア枠」モデルのトラック名をマウスでクリックして選択します。
- 2) 選択した「ドア枠」のトラック名でマウスの右ボタンを押します。表示されるプルダウンメニューから「別スコアへ移動」を選びます。



- 3) 「別のスコアへ移動」ダイアログボックスが表示されますので、「(新規スコア)」を選択し「OK」を押します。

プロジェクト内に、新しいスコアが追加され、その中にドアモデルトラックが移動しました。このスコアウィンドウには現在「スコア0」という名前が付けられています。この名前を変更してみましょう。

■スコアウィンドウの名前を変更する

次の手順を行ってください。

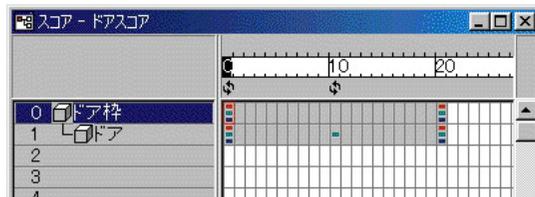
- 1) メニューの「表示」から「スコア一覧」を選びます。
- 2) 「スコアの表示」ダイアログボックスが表示されますので、スコア0スコアウィンドをクリックして選択します。
- 3) 一番上に表示された「スコア0」の名前を「ドアスコア」に書き換えて、キーボードの「Enter」を押します。最後に「Close」を押します。

ドアモデルの登録されたスコアウィンドウが、「ドアスコア」という名前に変更されました。それでは、ドアスコアウィンドウにループ枠を作成しておきましょう。

■ループ枠を設定する

次の手順を行ってください。

- 1) ドアスコアウィンドウの、ループ枠を設定するトラックの0フレーム目と10フレーム目をダブルクリックします。
(下の図を参考にしてください。)

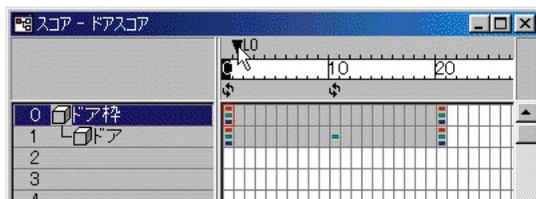


2つのループ枠が設定されました。続いて、ドアスコアウィンドウにラベルを設定しておきます。

■ラベルを設定する

次の手順を行ってください。

- 1) ドアスコアウィンドウの、ラベルを設定するトラックの1フレーム目を、マウスでダブルクリックします。ダブルクリックしたフレーム上にラベルが作成されます。



- 2) 続いて、作成された三角形のラベルマークをマウスでクリックします。「L0」とある名前を、「ドア開く」に書き換えて、キーボードの「Enter」を押します。

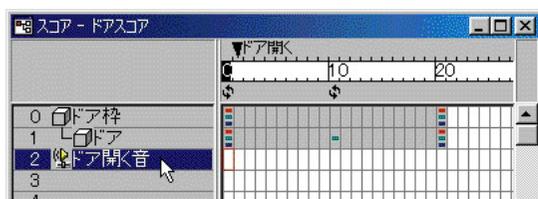
後から作成するスクリプトプログラムで使用するラベルを設定しておきました。

このスコアウィンドウの新規トラックに、再度「ドアの開く音」のサウンドキャストを登録しておきましょう。

このプロジェクトには、すでに「ドア開く音」という Wave サウンドキャストが用意されています。

キャストウィンドウの Wave サウンドキャストリストから、このスコアウィンドウの新規トラックにドラッグ&ドロップして登録してください。

「情報」ダイアログボックスが表示されますので、「はい」を選択して下さい。



次にトラックに直接登録された Wave サウンドキャストの、音を再生するフレームを指定します。

■ WAVE サウンドキャストの再生フレームを指定する

次の手順を行ってください。

- 1) ドアスコアウィンドウの、「ドア開く音」サウンドキャストが登録されたトラックの、1 フレーム目をマウスでダブルクリックします。

「再生制御」ダイアログボックスが表示されますので「再生制御」、「Play」をチェックし、「Ok」を押します。これでこのフレームを再生する際に、「ドア開く音」キャストのサウンドが再生されます。

現在このドアモデルは、3次元ワールドの中心にあります。ドアモデルの位置を少し変更してみましょう。

■ 3次元ワールドのモデルを移動する

次の手順を行ってください。

- 1) 親スコアであるルートスコアウィンドウの、「ドアスコア」トラックの0 フレーム目をクリックして選択します。



- 2) ステータスウィンドウで、選択している 0 フレーム目のキーフレームを、座標値 (Position) Z = 「330」に書き換え、キーボードの「Enter」を押します。



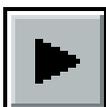
- 3) 続いて、ルートスコアウィンドウの「ドアスコア」トラックの 20 フレーム目のキーフレームをクリックして選択します。



- 4) 同じく、ステータスウィンドウで、選択している 20 フレーム目のキーフレームを、座標値 (Position) Z = 「330」に書き換え、キーボードの「Enter」を押します。



初期化



再生

「ドアスコア」トラックの座標値を変更して、ドアモデルの位置を変更しました。

ここで一度、プロジェクトを再生してください。

キーボードのカーソルキー「←→↑↓」を使用して、ウォーカーというモデルキャラクターを操作します。ドアモデルのある場所まで移動して、ドアモデルとウォーカーを重ねてみてください。現在は「当り判定」命令を使用したスクリプトを設定していませんので、ドアが開くアニメーションを行うことはありません。

当り判定と条件分岐命令を使用する

それでは、モデル同士の「当り判定」を使用したスクリプトキャストを作成してみましょう。

当り判定と条件分岐

作成するスクリプトは、「スクリプト」トラック設定しますので、はじめにドアスコアウィンドウにスクリプトトラックを作成します。

■スクリプトトラックを作成する

次の手順を行ってください。

- 1) ドアスコアウィンドウの、トラック名が表示されている所にカーソルを移動して、マウスの右ボタンを押します。
(ドアスコアウィンドウが表示されていない場合、ルートスコアウィンドウの「ドアスコア」トラックをダブルクリックしてください。)
- 2) 表示されるプルダウンメニューから「ベーストラックの新規作成／スクリプト」を選びます。



「ドアスコア」ウィンドウの新規トラックに、「スクリプト」トラックが作成されました。スクリプトトラックの0フレーム目に、新しいスクリプトキャストを設定してみましょう。

■スクリプトトラックにスクリプトを登録する

次の手順を行ってください。

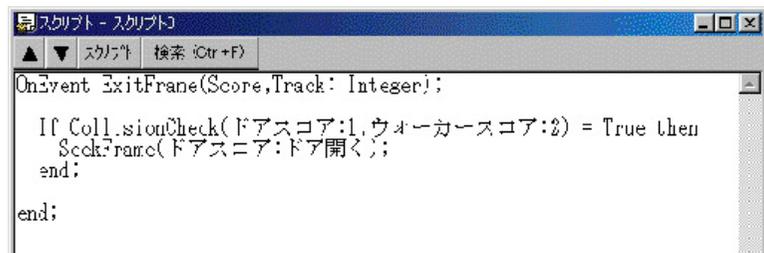
- 1) ドアスコアウィンドウの、スクリプトトラックの、0フレーム目をマウスでダブルクリックします。



- 2) ダブルクリックしたフレームに、スクリプトキャストが設定されます。同時に、設定されたスクリプトキャストのプロパティウィンドウが表示されます。
- 3) 開かれたプロパティウィンドウに書かれているプログラムの1行目と2行目の間に、次のプログラムを入力します。

```
If CollisionCheck( ドアスコア :1, ウォーカースコア :2)
= True then
    SeekFrame( ドアスコア : ドア開く );
end;
```

1行目と2行目の間に、上のテキストを入力して、スクリプトキャストを下の図のように書き換えました。

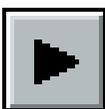


このプログラム命令の意味を説明します。

『このスクリプトを設定したフレームを再生中に、もし、ドアスコアウィンドウの1番のトラックに登録されたモデルと、ウォーカースコアウィンドウの2番のトラックに登録されたモデルが重なったならば、ドアスコアウィンドウの再生フレームを、「ドア開く」ラベルの付いたフレームへジャンプさせなさい。』という内容です。



初期化



再生

それでは、プロジェクトを再生してください。制御パネルウィンドウの「初期化」ボタンを押してから、「再生」ボタンを押してください。

ウォーカーというモデルキャラクターを操作して、再度、ドアモデルと重ねてみてください。今度は「当り判定」命令を使用したスクリプトが設定されていますので、ドアが開くアニメーションを行いました。

また再生画面の中には、カギのモデルが表示されていると思います。このカギモデルにも、スクリプトの「当り判定」命令が使用されています。ウォーカーがカギモデルと重なると、カギモデルを非表示にし、同時に画面上にカギのビットマップキャストを配置するようにスクリプトが設定されています。

今度は、このドアモデルがドアを開くアニメーションを行うまでに、もう1つ条件を付けてみましょう。

ウォーカーが、ドアモデルと重なっても、その前にカギモデルを取っていないければ、ドアが開くアニメーションを行わないようにしてみます。

まず、プロジェクトの中に、「変数」という「値をいれておく入れ物」のようなものを追加登録します。

■スクリプトキャストを変更する

次の手順を行ってください。

- 1) キャストウィンドウのリストを、スクリプトキャストリストに切り替えるために、キャストウィンドウ上部にある「script」と書かれたタブをマウスでクリックします。



- 2) スクリプトキャストリストの0番に登録された、「初期化」というキャスト名をマウスでダブルクリックします。
- 3) 「初期化」スクリプトキャストのプロパティウィンドウが表示されます。

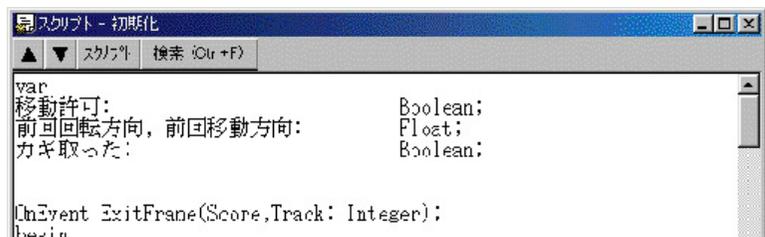
「初期化」というスクリプトキャストには、すでにスクリプト命令が書かれています。後半のスクリプト入門の項で説明いたしますので、ここではその説明を省略します。

このスクリプトの中にプログラムを追加します。

- 4) 開かれたプロパティウィンドウに書かれているプログラムの3行目の下に、次のプログラムを入力します。

カギ取った : Boolean;

上のテキストを入力して、スクリプトキャストプログラムを次の図のように書き換えました。



この追加入力したプログラム命令の意味を説明します。

『プロジェクトの中に、0か1しか入らない「カギ取った」という名前の「値をいれておく入れ物（変数）」を追加登録する』という内容です。

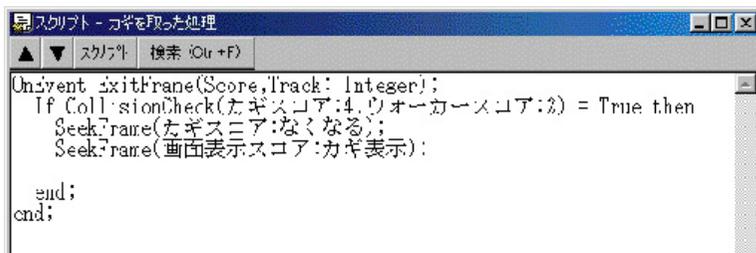
続いて、カギモデルが、ウォーカーと重なった時に使用されるスクリプトキャストに、再度追加命令を書き加えましょう。

■スクリプトキャストを変更する

次の手順を行ってください。

- 1) キャストウィンドウのスクリプトキャストリストで、「カギを取った処理」というキャスト名をマウスでダブルクリックします。
- 2) 「カギを取った処理」スクリプトキャストのプロパティウィンドウが表示されます。

「カギを取った処理」というスクリプトキャストにも、すでにスクリプト命令が書かれています。



```

OnEvent ExitFrame(Score,Track: Integer);
If CollisionCheck(カギスコア:4,ウオーカースコア:2) = True then
    SeekFrame(カギスコア:なくなる);
    SeekFrame(画面表示スコア:カギ表示);
end;
end;

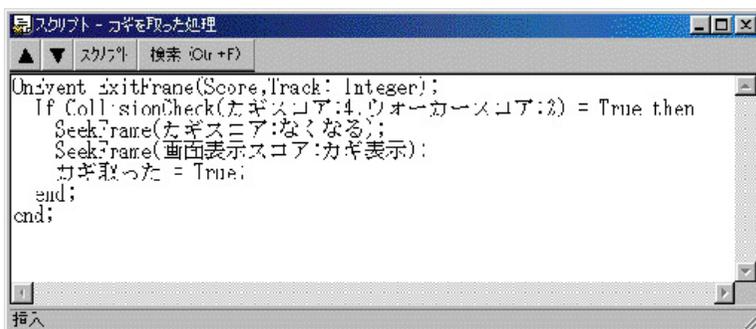
```

このスクリプトの中にプログラムを追加しましょう。

- 開かれたプロパティウィンドウに書かれているプログラムの5行目に、次のプログラムを入力します。

カギ取った = True;

上のテキストを入力して、スクリプトキャストを下の図のように書き換えました。



```

OnEvent ExitFrame(Score,Track: Integer);
If CollisionCheck(カギスコア:4,ウオーカースコア:2) = True then
    SeekFrame(カギスコア:なくなる);
    SeekFrame(画面表示スコア:カギ表示);
    カギ取った = True;
end;
end;

```

この変更されたプログラム命令の意味を説明します。

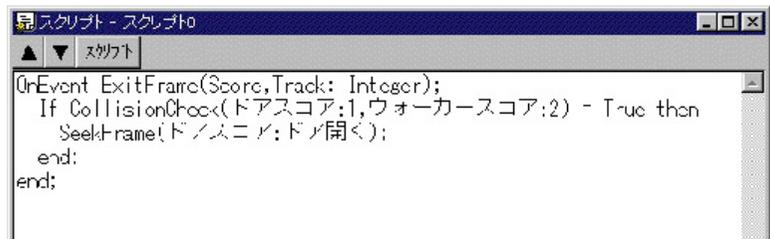
「このスクリプトを設定したフレームの再生中に、もし、カギスコアウィンドウの4番のトラックに登録されたモデルと、ウオーカースコアウィンドウの2番のトラックに登録されたモデルが重なったならば、ドアスコアウィンドウの再生フレームを、「ドア開く」ラベルの付いたフレームへジャンプさせて、「カギ取った」という名前の変数の中に、1 (True) を入れなさい。」という内容です。

ここで、今までの作業の流れを説明しますと、プロジェクトが再生された時に、「カギ取った」という名前の変数が作成されるようにして、ウォーカーとカギモデルが重なった時に、その「カギ取った」変数の中の値を0から1に変わるように設定しました。最後に、ウォーカーとドアモデルが重なった時のスクリプトキャストに変更を加えます。

■スクリプトキャストを変更する

次の手順を行ってください。

- 1) キャストウィンドウのスクリプトキャストリストで、「スクリプト0」というキャスト名をマウスでダブルクリックします。
- 2) 「スクリプト0」スクリプトキャストのプロパティウィンドウが表示されます。



このスクリプトキャストは、以前の作業で、ドアモデルの登録されている「ドアスコア」に設定したスクリプトキャストです。

このプログラム命令の意味を、再度説明しますと、次のような内容になっています。

『このスクリプトを設定したフレームを再生中に、もしドアスコアウィンドウの1番のトラックに登録されたモデルと、ウォークースコアウィンドウの2番のトラックに登録されたモデルが重なっていたならば、ドアスコアウィンドウの再生フレームを、「ドア開く」ラベルの付いたフレームへジャンプさせなさい。』

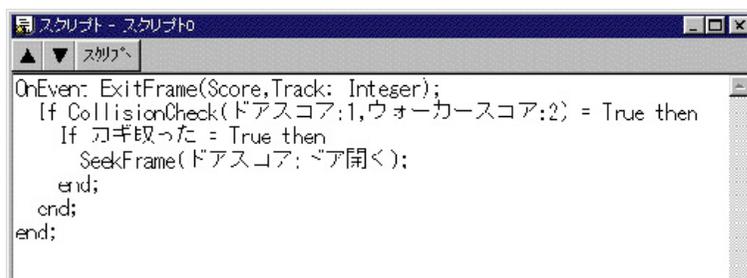
このスクリプトプログラムを、次のように書き換えてください。

```

OnEvent ExitFrame(Score,Track: Integer);
  If CollisionCheck( ドアスコア :1, ウォーカースコア :2)
  = True then
    If カギ取った = True then
      SeekFrame( ドアスコア : ドア開く );
    end;
  end;
end;

```

このようにテキストを書き換えて、スクリプトキャストを下の図のように変更しました。



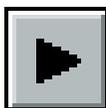
この書き換えられたプログラム命令の意味を説明します。

『このスクリプトを設定したフレームを再生中に、もしドアスコアウィンドウの1番のトラックに登録されたモデルと、ウォーカースコアウィンドウの2番のトラックに登録されたモデルが重なっていて、さらに、「カギ取った」という名前の変数の中が、1 (True) であったならば、ドアスコアウィンドウの再生フレームを、「ドア開く」ラベルの付いたフレームへジャンプさせなさい。』という内容です。

それでは、最後にプロジェクトを再生してみてください。



初期化



再生

DigitalLoca チュートリアルガイド終了！

…本当にお疲れさまでした！これで、DigitalLoca チュートリアルガイドは終了です。

今回作成した作品には、バックミュージックの設定や、イベント命令の種類等、まだまだ未完成な部分がいくつか存在します。

今回作成した作品に、もう少し手を加えたプロジェクトデータが、DigitalLoca のインストールされたディレクトリの中の、「Tutorial¥Walker¥Completion」フォルダに入っていますので、参考までにご覧ください。

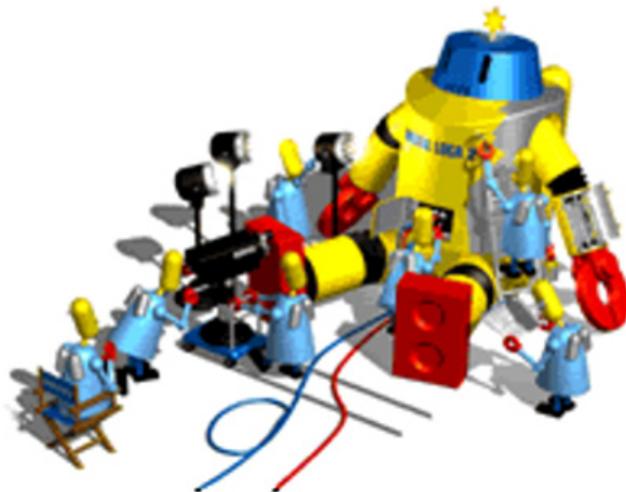
今回制作した作品は、DigitalLoca で制作することのできる作品の、ほんの一例でしかありませんが、これから皆様が作品を制作される上での参考になれば幸いに思います。

あなたの想像力と表現力を活かして、楽しい作品をどんどん制作してみてください。



DigitalLoca3 for Stylos

スクリプト入門



第1章：デジタルロケにおけるスクリプト

・はじめに

最近、「マルチメディアコンテンツ」という言葉をよく耳にします。「マルチメディアコンテンツ」とは、「2Dや3Dといった画像、テキスト、効果音やBGMとなる音声やビデオ動画など様々な(マルチ)な素材(メディア)から構成された作品(コンテンツ)」のことです。最も皆さんが馴染み深い「マルチメディアコンテンツ」の代表といえば「ゲーム」ではないでしょうか。デジタルロケはそんなマルチメディアコンテンツを制作するためのツールです。

マルチメディアコンテンツに求められる重要な要素の一つにインタラクティブ性(双方向性)があります。つまりユーザーの何らかのアクション(マウスクリックやキー入力など)に対応して画面やストーリーが変化して行く事です。これは、映画などの一方方向のメディアでは得られない大きな魅力となります。

ところで、デジタルロケでは、こういったインタラクティブな処理をコンテンツに盛り込むために、スクリプト(簡易プログラミング言語)が用意されています。本編では、全くプログラミング経験の無い方を対象に、デジタルロケのスクリプトの基本を学習して行きます。最終的には、前編までで作成したコンテンツのスクリプトが全て理解出来る様になっています。本編を読み終えた頃には、あなたの作成したキャラクタを、自由自在に動かす事が出来る様になっているでしょう。

・学習を進めていく上で

何でもそうですが、スクリプトを学習して行く際にも、実際に自分で書いてみて、その結果を確認するという作業が非常に重要となります。ですから、ガイドを読んで理解できたら、次はその一部を変更して、結果として現れる変化を確認しましょう。結果が絵として目に見える分、デジタルロケにおけるこの作業は楽しいものです。試行錯誤を繰り返し、理解を深めていってください。

1-1. スクリプトとは

それではスクリプトがコンテンツの中でどのように働いているか見てみましょう。デジタルロケのチュートリアルフォルダから「¥Samples¥DragonBook.LCP」を開いてください。これはドラゴンという架空生物の特徴を紹介するコンテンツです。

このプロジェクトファイル(LCPファイル)を実行してみると、まずロゴが表示され、続いてコンテンツに関する説明のテキストが現れます。ここで画面内をマウスでクリックするとドラゴンが登場します。このドラゴンの翼や頭、さらには部屋の柱をクリックすると、それらのアクションに対応したアニメーションが始まります。ユーザーの働きかけ(マウスクリックなど)に応じて適切な絵やアニメーションを表示する、このインタラクティブ性(双方向性)こそ、マルチメディアコンテンツでは重要な性質であり、これを可能にしているのがスクリプトなのです。マウスでクリックしてもキーを押しても、それらに無関係で画面表示が続くのであれば、それは単なるムービーでしかありません。マルチメディアコンテンツがマルチメディアコンテンツたる所以は、まさにスクリプトにあるのです。

1-2. イベントとは

それでは実際にスクリプトを見てみましょう。まず、「キャストウインドウ」から「Script」タグを選択してください。ここに、このコンテンツで使用されている全てのスクリプトが登録されています。上から2番目の「スクリプト1」をダブルクリックしてみましょう。

```
OnEvent CastClick(Score,Track,Button,X,Y:Integer);
  SeekFrame(ロアニメーション);
end;
```

スクリプトキャストの編集画面が立ち上がり、「スクリプト1」の内容が表示されます。これは、ドラゴンが直立状態のアニメーションのときに、頭をクリックすると火を吹くアニメーションを始めるようにするためのスクリプトです。この3行からなるスクリプトを日本語で言いなおすと、

```
このキャストがクリックされた時(OnEvent CastClick)に、ルートスコアでの再生ヘッドの位置をラベル「ロアニメーション」の位置まで移動(SeekFrame(ロアニメーション))しなさい
```

という意味です。デジタルロケのスクリプトは「○○が起きたら××する」というように、何かしらのイベントに応じて処理が行われる仕組みとなっております。一般的には「イベント駆動型」と呼ばれるものです。ですから、デジタルロケのスクリプトキャストはどんなに長いものでも、

```
OnEvent イベント名
[begin]
処理
end;
```

という枠にそったものになっているはずですが、「処理」の部分が大きいだけで基本は変わりません。※この形をとらないもの（関数、手続きなど）もありますが、基本はこの「OnEvent」です。

この枠から外れて書いた場合、例えば、

```
OnEvent CastClick(Score,Track,Button,X,Y: Integer);
begin
end;
SeekFrame(ロアニメーション);
```

のように書いたものは、エラーとなってしまいます。最初、処理をどこに書けば良いかわからない事があるかもしれませんが、必ず「begin」と「end」の間に書くんだということを忘れないでください。

デジタルロケには Table1 に示すような9種類のイベントがあります。

DigitalLoca チュートリアル

Tablet デジタルロケのイベント一覧

イベント名	発生する内容
EnterFrame	新しいフレームが表示される直前
CastClick	ステージに表示されているキャストがクリックされたとき
MouseDown	マウスのボタンが押されたとき
MouseMove	マウスが移動したとき
MouseUp	押されているマウスのボタンが離されたとき
KeyDown	キーボードのキーが押されたとき
KeyPress	キーボードのキーが押されているとき
KeyUp	押されているキーボードのキーが離されたとき
ExitFrame	フレームの表示が終了したとき

いずれも、これから先にスクリプト作業を進めて行けば何度もお目に掛かるものばかりです。今、これらを知る必要は全くありません。ただ、デジタルロケでコンテンツを作る上で何かインタラクティブな要素を発生させたい時、それらの要素はこれらのイベントのタイミングで発生させる事が出来るのだという事だけ覚えておいてください。

1-3. スコアとの関係

以上で「このキャストがクリックされたとき～」の「クリックされたとき」というのは理解できたと思います。でも、「このキャスト」つまり「ドラゴンの頭」の指定はどこに書かれているんだ?という疑問が出てきます。しかしながら、これはいくらスクリプトを見ても分かりません。この疑問を解決するにはスコアを見なくてはなりません。

デジタルロケのスクリプトはキャストを作成しただけでは意味がありません。作成したスクリプトキャストをスコアに登録し、適切なイベントが発生して、始めて処理が行われるのです。そこで「スクリプト 1」がスコアのどこに登録されているか探してみましょう。

まず、スクリプトキャストを探しやすいように、スコアの表示モードを変更します。ルートスコアにおいてスコアのトラック名が書かれている部分を右クリックして、ポップアップメニューを開いてください。その中に「表示モード切り替え」というメニューがあります。ここで確認できるように、スコアには3種類の表示方法があります。初期状態ではキーフレームの情報が表示される「スコア」に設定されていますので、これを「スコア+スクリプト」に変更してください。そうすると、1つのトラックが2段になり、上段にキーフレーム情報、下段に登録されているスクリプトのキャスト番号が表示されるようになります。ここでトラック番号の25、26番を見てください。9～38フレームに先程の「スクリプト 1」のスクリプトキャスト番号「01」が登録されています。これが「このキャストがクリックされたとき～」の「このキャスト」に当たるわけです。つまりドラゴンの顔、顎がクリックされたときに「スクリプト 1」が実行されるのです。このようにデジタルロケのスクリプトは、その記述とスコアへの配置の両者によって完結するのです。スクリプトのスコアへの配置方法は全部で以下の3種類があります。

- 1.スコア上のスクリプトトラックに割り当てる方法
- 2.スコア上のスクリプトトラック以外のトラックに割り当てる方法
- 3.1つのスクリプトキャスト専用のトラックを作成し、そこに割り当てる方法

今回の「スクリプト 1」は 2 に相当します。また、ルートスコアの 0 番トラック [スクリプト] が 3 のスクリプトキャスト専用トラックです。1 の方法は今回は出てきませんが、他の 3D モデルキャストと同様に扱う方法です。

1-4. キャスト名

スクリプトキャストがスコア上のどこに配置されているかという情報は非常に重要です。しかし、大きなコンテンツを作成したとき、または何人かで作業をしたときは、どこに配置されているのかわからなくなるといった事も多々あります。そこで、そういった状態を回避するため、スクリプトキャスト名の付け方に工夫をしてみましょう。

今回は「スクリプト 1」となっていますが、例えば「配置してあるスコア名+処理内容」などのように一定の命名規則を決めて名前を付けると作業が楽になります。「スクリプト 1」の場合は「RootScore_FireAnim」のように命名しておけば、「このスクリプトはルートスコアに置かれていて火を吹くアニメーションに関係しているな」と第三者でも連想できます。

1-5. 処理

最後は、最も重要な「処理」の部分です。

```
SeekFrame(ロアニメーション);
```

これは

```
ルートスコアでの再生ヘッドの位置をスコア上のラベル「ロアニメーション」の位置まで移動しなさい。
```

という意味です。この SeekFrame 命令は、

```
SeekFrame([スコア名:]ラベル名);
```

と書く決まりになっています。スコア名が省略された場合には自動的にルートスコアが選択されます。上記の命令を省略せずに書くと、

```
SeekFrame(ルートスコア:ロアニメーション);
```

となります。SeekFrame 命令によってアニメーションの進行を自在に変化させることができるわけです。

・章末

以上が、最も簡単なインタラクティブ処理といえます。しかし、このたった 3 行で、「マウスでドラゴンの頭をクリックしたら火を吹く」といった処理が可能になりますし、これを膨らませていけばゲームも出来上がるのです。

DigitalLoca チュートリアル

次章では、スクリプトの概要及び決まり事を学習していきます。難しい用語もたくさん出てきますが、恐れずに頑張って行きましょう。

第2章：スクリプトの基礎

・はじめに

デジタルロケのスクリプトは非常に豊富な命令群を持ち、柔軟なプログラミングが可能となっております。これにより本格的な3Dゲームを作ることにも可能なわけですが、その反面本格的なプログラミングに近い作業を要するといった性質も持ち合わせております。そのためプログラミング経験の全く無い人にとってはスクリプト作業が一つの大きな壁となってしまいます。

そういった方がよく嘆いているのは、「型とか変数って何?」であるとか「手続き、関数って何だろう?」といった事です。確かにスクリプトを始めて、いきなり「変数の型には整数型・実数型・文字列型・論理型があります」などと言われても訳がわかりません。それが普通です。ですが、その内、「何でこんな事がわからなかったんだ」と言う事に必ずなりますので、最初は意味など考えないで、「こういうものだ」と思っただけければ結構です。

・章構成

本章では、スクリプトの基本構成要素について解説していきます。概念的な話が中心となりますが、前述のような事を踏まえて頑張ってください。本章で学習する内容は以下の通りです。

- ・変数, 定数
- ・型
- ・グローバル, ローカル
- ・関数, 手続き

2-1. 変数・定数

変数は変更できる値を表すものです。わかりやすく言うと、情報を入れて置くことのできる入れ物といえます。入れ物は変わらずに中の情報だけを変更することができます。作品の再生中に変更が生じる情報を管理するために使用します。例えば、ゲームの得点情報やプレイヤーの残り数を保持して置く場合などです。

これに対して、定数は変更のできない一定の値を表すものです。変更ができないという点を除けば変数と同じで、読み出し専用の変数とも言えます。作品の再生中に一定の情報を管理するために使用します。

2-2. 型

型とはその名の通り、変数や定数がどのような値を取ることが出来るかを示す物です。言い換えれば入れ物の種類です。デジタルロケのスク립トには Table2 に示す 4 種類の型があります。

Table2 デジタルロケで用意されている型

型	内容	説明
Integer	整数型	-2147483648～2147483647までの整数
Float	実数型	±5.0e-324～1.7e308までの実数
String	文字列型	自由な長さの文字列
Boolean	論理型	TrueまたはFalse

ある入れ物には、それに合った形式の情報しか入りません。ですから Integer 型の変数には、整数の情報しか入れる事ができません。Integer 型の変数に、「110」などと実数型の値を入れようとすると、「Error: ~型が違う」といったエラーになってしまいます。

2-3. 変数・定数を使うには

変数と定数を使用するためには、あらかじめ型を定義しておく必要があります。つまり、値の入れ物をあらかじめ用意する必要があるという事です。変数の定義に関して表記は次のようになります。

```
変数名 [, 変数名]:型 [=初期化値];
```

これを予約語「var」の後に記述します。

例えば、整数の値を取る変数「IntA」を宣言するには、

```
var  
IntA : Integer ;
```

と記述すれば良いわけです。しかし、この状態だと入れ物が用意されただけで、中には何も有効な情報は入っていません。正確には宣言された時点で情報はありますが、プログラマーが意図的に用意した情報ではないという意味で「有効でない」というわけです。宣言後に値を入れてもかまいませんが、宣言と同時に値を代入することも可能で、この値を「初期化値」といいます。初期化値を 10 とするには、以下のように記述します。

```
var  
IntA : Integer =10;
```

こうすると、「IntA」という変数が用意されると同時に、その中には整数の「10」が入る事になります。宣言以降は、

```
IntA=11;
```

のように、整数情報だったら何を入れても良くなります。また、同じ型の変数なら、変数名を「,」で区切って続けて記述する事により、同時に複数個を宣言する事が出来ます。例を挙げると、

```
var
  Str1:String;
  Str2:String;
  Str3:String;
```

という場合に、

```
var
  Str1,Str2,Str3:String;
```

と1行で記述する事が出来ます。定数の場合も変数の時とほぼ同様の記述で宣言できます。ただし、

1. 予約語は「var」ではなく「const」となる。
2. 初期化値を省略することはできない。
3. 同じ型でも複数個を同時に宣言することはできない。

という点で異なります。ですから、整数の値を取り初期化値「20」の定数「IntB」を宣言するには、

```
const
  IntB : Integer =20;
```

と記述することになります。また、変数の場合と異なり、宣言後、

```
const
  IntB : Integer =20;
```

のように値を代入することはできません。

以上、変数と定数の宣言の仕方について学習してきましたが、実際に作品内で使用するにはどこで宣言すればよいのでしょうか?ここで問題となってくるのが「グローバルとローカル」という概念です。

2-4. グローバルとローカル

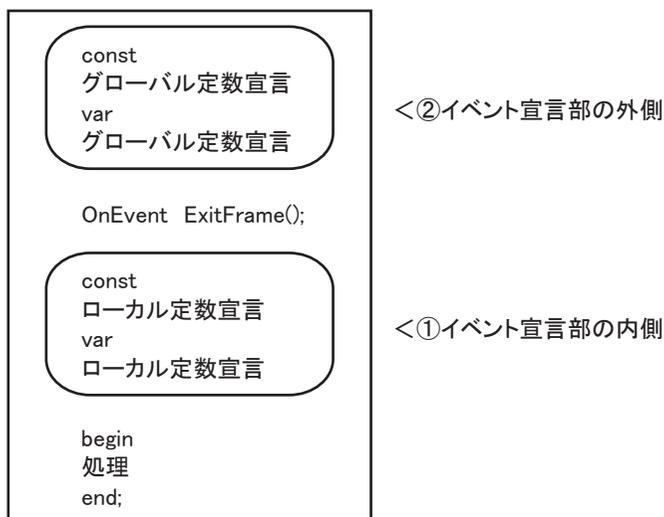
以前、「処理は必ず begin と end の間に書く」と言いました。これに対し、変数と定数の宣言は書く場所が違います。その場所は、

DigitalLoca チュートリアル

- 1. イベント宣言部の内側
- 2. イベント宣言部の外側

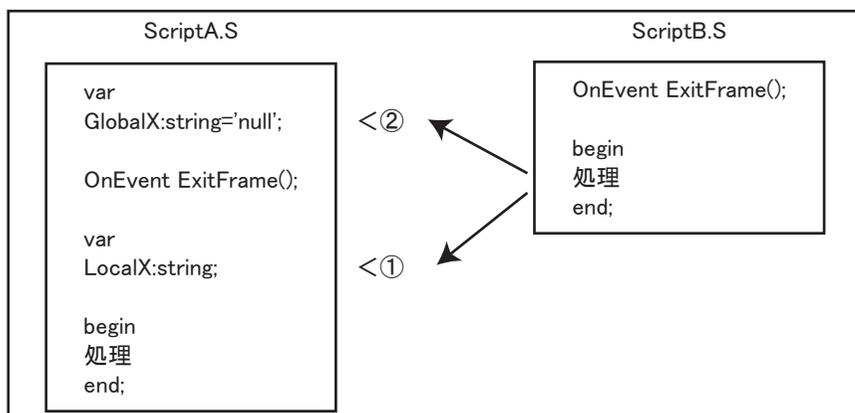
の2箇所になります。そして、ここが一番大事なのですが、その2箇所の内どちらで宣言するかによって、変数・定数の性質が異なってくるのです。1で宣言されたものは「ローカル変数 / 定数」、2で宣言されたものは「グローバル変数 / 定数」と呼ばれます。(Fig1 参照)

Fig1 ローカルとグローバル



それでは、どのように性質が変わってくるのでしょうか?実は宣言したはずの変数が見えなくなってしまう事があるのです。まずは Fig2 を見てください。

Fig2 宣言場所による違い



このように ScriptA.S、ScriptB.S の2つのスクリプトソースがあり、ScriptA.S の中には、文字列型のグローバル変数「GlobalX」、ローカル変数「LocalX」が宣言されているとします。この場合、ScriptB.S からだと、ScriptA.S

の②の部分の変数 / 定数は見ることができますが、ScriptA.Sの①の部分の変数 / 定数は見えないのです。ですから ScriptB.S の処理の部分に、

```
LocalX="TEST";
```

のようなスクリプトを書いて、作品を再生しようとする、ScriptBにおいて「未定義の識別名です」というエラーが発生してしまいます。これは、ScriptBが「LocalXなんて変数は知らないよ」と言っているのです。今度は、処理の部分の内容を、

```
GlobalX="TEST";
```

としても、エラーは発生しませんし、GlobalXの中は TEST という文字列が代入されます。このように、変数 / 定数はその宣言場所によってグローバル・ローカルという性質を持ちます。ですから必要に応じて、変数が定数か、グローバルかローカルかを使い分けなければなりません。

・メモ

「別に使い分けなくても全部グローバルで書けば良いじゃないか」と思われるかもしれませんが、きちんとローカル変数 / 定数を使い分ける事は良い事なのです。その理由を1つあげると「メモリを浪費しない」という点があります。変数・定数というのは宣言されると、コンピュータのメインメモリ上のある領域を確保し、そこに情報を書きこんでいきます。つまり、このメインメモリ上のある領域というのが情報の入れ物となる訳です。そして、グローバル変数 / 定数の場合は、作品の再生が始まってから終わるまで、ずっとその領域を確保します。これに対しローカル変数 / 定数の場合、必要になった時(宣言してあるイベントなどが実行される時)に始めて領域が確保され、不要になると(そのイベントなどでの処理が終了したとき)その領域は破棄されます。ですから、全てをグローバル変数 / 定数で宣言すると、不要になった入れ物も無駄にメモリに置かれたままになります。メモリの大容量化が進んだとはいえ、限りある貴重なメモリを浪費するのはもったいない話です。グローバルとローカルはきちんと使い分けましょう。

2-5. 手続き

堅い話ばかりで、もう嫌気がさしてる方もいらっしゃるかもしれませんが、あと少しの辛抱です。この章の最後の項目である「関数 / 手続き」を学習していきましょう。

まず、一言で「関数 / 手続き」を説明しますと、

```
繰り返し登場する一連の処理を一まとめにしたもの
```

と言えます。しかし、この一言で理解するのは難しいので、具体例を見ながら、学習していきましょう。

なお、以下の例のスクリプト、SampleA・Bは、ただ「関数・手続き」というものを説明するためのものであり

DigitalLoca チュートリアル

特に意味はありません。以下の2つのスクリプトを、次のようにして、それぞれの動きを確認してください。

1. 新規プロジェクトを立ち上げる。
2. スクリプトキャストを新規作成する。
3. 書かれているスクリプトを打ち込む。
4. スクリプトキャストをスコアに登録し、0フレームに当たるキーフレームに設定する。

実行してみると両者とも、3回メッセージボックスが表示されます。その中にある数字は「1」、「3」、「6」となっているはずですが、この2つのスクリプトは全く同様に動きますが、その書かれ方は異なります。この違いが手続きを使用しない場合と、を使用した場合の違いなのです。

<SampleA>

```
var
  i: integer=0;

OnEvent ExitFrame(Score,Track: Integer);

begin

  i=i+1;          ...①
  MessageBox(IntToStr(i)); ...②
  i=i+2;          ...③
  MessageBox(IntToStr(i)); ...④
  i=i+3;          ...⑤
  MessageBox(IntToStr(i)); ...⑥

end;
```

まず、SampleAをじっくり見てみましょう。最初に、整数型のグローバル変数*i*が宣言され、初期値0が代入されています。次に、このスクリプトキャストが置かれているフレームを出る時に行われる処理が書かれています。行われている6行の処理を順に見て行くと、

- ① 変数*i*に1を足す。
- ② 変数*i*の中身をMessageBoxを使用して表示する。
- ③ 変数*i*に2を足す。
- ④ 変数*i*の中身をMessageBoxを使用して表示する。
- ⑤ 変数*i*に3を足す。
- ⑥ 変数*i*の中身をMessageBoxを使用して表示する。

※ MessageBox 命令で表示できるのは文字列データだけです。そこで、整数である変数*i*の値を IntToStr 命令によって文字列に変換した後、MessageBox 命令に渡しています。

という処理を行っています。

ここで何か気付きませんか?①③⑤といった奇数行では変数*i*に整数値を足すという非常に似た処理が行われており、②④⑥といった偶数行では変数*i*の中身を MessageBox に表示するという全く同じ処理が行われてい

ます。言い換えると、この6行では「変数*i*に整数値を加算し、その中身を表示する」という処理を3回繰り返しているのです。これが冒頭で「関数・手続き」を一言で説明したときに出てきた「繰り返し登場する一連の処理」なのです。そして、これらをまとめて手続きとしたものが SampleB なのです。

```
<SampleB>

var
  i: integer=0;

procedure syori(num:integer);

begin
  i=i+num;           ...p①
  MessageBox(IntToStr(i)); ...p②
end;

OnEvent ExitFrame(Score,Track: Integer);

begin
  syori(1);          ...e①
  syori(2);          ...e②
  syori(3);          ...e③
end;
```

SampleB を見ていきましょう。先程の SampleA では ExitFrame イベントが一つあるだけでした。これに対し、SampleB には ExitFrame イベント以外に、procedure という記述があります。これが「手続き」と呼ばれるものです。“syori” という名前がこの手続きでは、繰り返し行われる「変数*i*に整数値を加算し、その中身を表示する」という処理を行っています。デジタルロケでは「手続き」は次のような構文に従って宣言します。

```
procedure 手続き名 [(引数名: 引数の型)];
```

これを SampleB の手続き “syori” と見比べてください。

```
procedure syori(num:integer);
```

“syori” が手続き名 (識別子)、 “num” が引数名、その型が “integer” となっています。このように宣言した後は、今までイベントでの処理を記述してきたのと同様、begin ~ end; 間、その手続きで行いたい処理を書けば良いのです。今回の syori 手続きには、

```
p①変数iにnumを加算する。
p②変数iの中身をMessageBoxを使用して表示する。
```

という処理が記述されています。

しかし、このままでは上記のような処理を行ってくれる手続きが宣言されただけで、実際には使われていま

DigitalLoca チュートリアル

せん。そこで、ExitFrame イベントでは、この宣言した syori 手続きを呼び出して、実際に処理をしています。ExitFrame イベントでは、

e①引数(numの値)を1としてsyoriを行う。
e②引数(numの値)を2としてsyoriを行う。
e③引数(numの値)を3としてsyoriを行う。

というように、syori 手続きを 3 回呼び出しています。つまり、procedure 部分で「変数 i に整数値の引数を加算し、その中身を表示する」手続きを宣言し、ExitFrame イベントでその手続きを 3 回実行している訳です。

SampleA と B の ExitFrame イベントを見比べてみてください。B の方が非常にすっきりしていて、ここで行われていることがはっきりと分かるのではないのでしょうか?それに比べると A の方は大きな流れが非常に見えにくくなっています。手続きを上手く使えば、メインのスク립トでは、そこで何をやりかえりかという大きな流れを記述するだけで済み、後々のメンテナンスを楽にすることができるのです。

2-6. 引数

ところで、手続きとはどういったものかを見てきたわけですが、1つ説明が不足している部分がありますね。そうです、「引数」です。先程は手続きというものを説明するため、見て見ぬふりをしましたが、ここでは、じっくり「引数」というものを学習しましょう。

引数とは、呼び出し側 (ExitFrame) から手続き側 (syori) に送られる情報と考えてください。厳密に言うと、もっと小難しい定義になるのですが、今はこのように考えておいて問題は無いと思います。先程の例で考えれば、syori 手続きでは、num という integer 型の引数を使用しています。syori 手続きは、呼び出されると、この num という引数に応じて処理を行います。そして、この num に入れられる数値は、呼び出し側が与えています。e①の「syori(1);」の()の中の値、つまり“1”が num に代入されるのです。ここで、引数の意味を理解するために、引数というものがない場合を考えてみましょう。

syori 手続きの前半の p①では、「変数 i に num を加算する」という処理を行っています。ここで引数 num を利用していますが、もし、この num が使えなかったらどうなるでしょう。最初、変数 i に足すのは 1 ですから、syori 手続きの p①「i=i+num;」を「i=i+1;」と書き換えることになるでしょう。しかし、これでは 2 を足すときには使えません。そうした場合、syori2 という別の名前で、新たに手続きを宣言する必要が出てきてしまいます。ですが、これではわざわざ手続きを作成する意味がありません。このように 3 個も手続きを用意するぐらいなら、SampleA のように書いてしまった方が良いでしょう。

これに対して、引数を使用すると、加算する数値が 1 でも 2 でも、「変数 i に何らかの整数値を加算する」という一文で、表すことができます。これでなんとなく引数というものの意味がわかったのではないのでしょうか? なんとなくで結構です。後は、実際に自分で手続きを作成していく中で理解できると思います。

2-7. 関数と戻り値

ここまででは「手続き」というものを学習してきました。残るは「関数」です。しかし、手続きを学習してしまった皆さんでしたら恐れることはありません。関数も手続き同様、一連の処理をまとめたものですが、手続きとの

違いは、戻り値を得ることができるという点です。手続きには引数を与えることによって、その引数に応じた処理をさせることができました。関数にも引数を与えることができますが、さらに、処理の結果などを呼び出し側に返すことができます。そして、この返ってくる値が戻り値なのです。それでは SampleC を見てください。これも、SampleA・B の時と同様、実際に打ち込んで、試してみてください。MessageBox が現れて “2” という値が表示されていれば問題ありません。

<SampleC>

<pre>function warizan(bunbo,bunsi:Integer):Integer; begin Result = bunsi mod bunbo; end; OnEvent ExitFrame(Score,Track:Integer); var amari:Integer; begin amari=warizan(3,11); MessageBox(IntToStr(amari)); end;</pre>	<p><f①></p> <p><e①></p> <p><e②></p>
--	---

まず、手続きの場合と同様、関数を宣言する際の構文を示します。

```
function 手続き名 [(引数名 : 引数の型)]: (戻り値の型);
```

これを SampleC の関数 “warizan” と見比べてください。

```
function warizan(bunbo,bunsi:integer) :integer;
```

“warizan” が関数名(識別子)、“bunbo” と “bunsi” が引数名、その型が “integer”、そして戻り値の型も “integer” となっています。手続きの場合と異なるのは、最初に「procedure」ではなく「function」と書き始める事、最後に戻り値の型を指定する事の2点だけです。「begin ~ end;」間に、この関数で行いたい処理を書けば良いのも手続きの時と一緒に。さてここで、「戻り値の型は決めたけど、名前はどうするの?」という疑問が出てきます。しかし、戻り値の名前は決める必要がありません。戻り値の名前は必ず「Result」と決まっているのです。ですから、呼び出し側に返してあげたい値は、「Result」という名前の変数に入れるだけで済むのです。

それでは SampleC の中身を見ていきましょう。まず、warizan 関数では、

f①引数bunboで引数bunsiを割り、その余りを戻り値とする。
 ※ □mod□は剰余演算子とよばれるもので、除算を行いその余りを求めます。
 (例:「7 mod 2」は7/2=3余り1なので「1」になります)

DigitalLoca チュートリアル

という処理を行っています。ただそれだけの処理です。ここで先程説明した「Result」が登場します。その後、今宣言した warizan 関数を ExitFrame イベントで呼び出しています。

e①'引数を3、11としてwarizan関数を行い、その戻り値を整数型変数amariに代入する。
e②'変数amariの中身をMessageBoxを使用して表示する。

この結果、「lmod3」の演算結果である2が変数 amari に代入され、MessageBox に表示されるのです。

どうでしたか?手続きが理解できていれば、関数の理解にもそれほど苦しまずに済んだのではないのでしょうか。

・章末

本章では少しデジタルロケから離れて、スクリプトの約束事を勉強しました。完全に理解できている必要はありません。「ちょっと不安は残るけど、大体は理解したつもり」という程度であれば十分です。自信を持って次の章に進みましょう。次章からはデジタルロケで頻繁に使用されるスクリプト命令を学習します。

第 3 章 : 代表的な命令

・はじめに

本章では、デジタルロケでよく使用される代表的な命令を学習していきます。ただ今回は、実際にサンプルコンテンツを見ながらスクリプト命令を紹介していきますので、前章のように堅い話ばかりではありません。ご安心ください。

・本章で学ぶこと

今回サンプルとして使用するのは「EscapeGuy」という 3D アクションゲームです。まず、デジタルロケのチュートリアルフォルダから「¥Tutorial¥Samples¥EscapeGLCP」を開いて実行してみてください。操作は以下のようになっています。

カーソルキー	主人公の移動
Spaceキー	攻撃(銃、パンチ)
Returnキー	ドアオープン

ゲームとしては、登場する敵キャラクター(ザコ2体、ボス1体)を倒しながら、ゴールを目指すという形式のものです。これから紹介するスクリプト命令を学習する前に、まず、どんなコンテンツか、実際に遊んで確認しておきましょう。

それでは、EscapeGuy の中で使用されている代表的な命令群について詳細に見ていきましょう。次章からは、そういった命令群をもとにどのようにアルゴリズムを組んでいるか(コンテンツを作成しているか)を解説していきますので、本章では先ず個々の命令をしっかり学習しましょう。本章で扱う命令群は以下のものです。

1. キー情報の取得
2. 条件分岐命令
3. パペット化
4. 衝突判定

3-1. キー情報の取得

デジタルロケの大きな魅力の1つはインタラクティブなコンテンツを作成できることです。ユーザーの操作に対応して画像が変化していくのですから、まず、ユーザーがどのような操作を行ったかを知る必要があります。

EscapeGuy ではユーザーがカーソルキーを押すと、それに対応して主人公キャラクター Guy が歩きます。それでは EscapeGuy の中では、どのようにしてカーソルキーに関する情報を取得しているのでしょうか? 「EscapeGLCP」の中のスクリプトキャスト 2 番「プレイヤースコア_移動操作」を開いてください。この上部(15~19行目)に、

DigitalLoca チュートリアル

```
(*キーの読み込み*)
if GetKeyState(VK↑UP) then 移動方向 = 移動方向 -1; end;
if GetKeyState(VK↓DOWN) then 移動方向 = 移動方向 +1; end;
if GetKeyState(VK←LEFT) then 回転方向 = 回転方向 -1; end;
if GetKeyState(VK→RIGHT) then 回転方向 = 回転方向 +1; end;
```

という記述があります。この部分では、カーソルキーが押されているか押されていないかというキー情報を取得し、それをもとにキャラクターの新しい位置情報を計算しようとしています。なお“移動方向=移動方向-1”というのは計算の準備です。この“GetKeyState”という命令がキー情報を取得するための命令です。“GetKeyState”は次のように宣言されている関数 (function) です。

```
function GetKeyState(Key:Integer):Boolean;
```

この関数は、

```
Keyという整数型の引数を受け取り、その引数の値をキーコードとするキーが
押されているか調べる。もし押されていたら論理型のTrueを、押されていないなら
Falseを戻り値として返す。
```

というものです。キーコードというのは、あらかじめキーボードのそれぞれのキーに割り当てられている番号のようなものです。キーコードの一覧表を見るには、メニューから「ヘルプ」-「トピックの検索」を実行します。次に、立ち上がったヘルプで検索キーワードに「キーコード」と入力すると、「キーコード定数」という項目が検索されます。まずはそちらを参照してください。

これらが、キーボードのどのキーに何番が割り当てられているかを示す一覧表なのです。このとき注意しなければならないのは、その番号が、日頃私達が慣れ親しんでいる10進数ではなく、16進数で割り当てられているということです。ですから“GetKeyState”命令の引数として、直接キーの番号を使用する際には、「この数字は16進数表記ですよ」ということを示すために番号の前に「\$」を付けなくてはなりません。例えば Shift キーの情報を取得するためには、

```
□GetKeyState(10)□…②
```

ではなく、

```
□GetKeyState($10)□…③
```

と書かなければなりません。②のように書くと、Shift キーではなく、ジョイスティックのボタンの状態を調べることになってしまいます(10進数表記の「10」は、16進数表記の「0A」になります)。③のように書くと、「\$」記号によって「10」が16進数として認識されますので、Shift キーの状態が調べられるのです。

しかし、「Shift キーのキーコードは \$10」というように全てのコードを覚えるわけはいいきませんし、いちいちキーコード表を見るのも面倒な話です。また、後からスクリプトを見直すときにもキーコード表のご厄介になってしまい

ます。そこで用意されているのが「キーコード定数」と言うものです。これは、既に定義されている定数というだけで、普通の定数と同様の働きを示します。キーコード定数を使うと、③は、

```
□GetKeyState(VK□SHIFT)□…④
```

と書き換えることができます。④も③と全く同様な処理を行います。また、こう書けば、「Shift キーの状態を調べている」ということが、キーコード表を見なくても分かります。①のスク립トはこのキーコード定数を利用して書かれており、カーソルキーの「↑」「↓」「←」「→」の状態を調べているのです。

3-2. 条件分岐命令

スクリプトを組む上で最も多く出てくる処理に「条件分岐」があります。つまり「もし、○○だったら××して、そうでなかったら△△しなさい」というように、プログラムの流れを制御するものです。先程の①には、

```
if GetKeyState(VK□UP) then 移動方向=移動方向-1; end;…⑤
```

という処理がありました。この内、“GetKeyState(VK_UP)” は、カーソルキー「↑」の状態を調査しているということ、先程説明した通りです。そして、キー状態の調査が終わると、“GetKeyState(VK_UP)” の部分には、戻り値である「true」か「false」が入っています。これでキーが押されているか、押されて無いか分かりました。残りの部分は、この情報をもとに処理を分岐させているのです。このようにプログラムの実行の流れを制御する命令を「ステートメント」といい、ここで使用されている if 文は以下のように構文に沿って使用されます。

```
if (条件式) then (処理1) [ else (処理2) ] end;
```

この if 文は、

```
(条件式)が満たされている時は、then以下の(処理1)に書かれている処理を行い、
(条件式)が満たされて無い時は、else以下の(処理2)を処理する。
else(処理2)が無い時は何もしない。
```

という命令です。ですから、⑤の例の各要素は次のようになります。

```
(条件式)   GetKeyState(VK□UP)
(処理1)   移動方向 = 移動方向 -1
※⑤では[ else (処理2) ]の部分がありません。
```

つまり

```
もしカーソルキーの「↑」が押されていたら「移動方向」という変数の中身を1減らしなさい。
```

DigitalLoca チュートリアル

という処理が行われているのです。なお、条件式の「GetKeyState(VK_UP)」は、実際は「GetKeyState(VK_UP)=True」と書くのですが、上記の記述でも同じ判定が出来ます。

3-3. パペット操作

デジタルロケの特徴的な機能にパペット機能があります。これはモデルの表示位置などを、スコアに書かれているデータではなく、スクリプトによって決定する為の機能です。この機能を先程の GetKeyState 命令や if 文と組み合わせることにより、キャラクターを操り人形のようにカーソルキーで操作することが可能となります。パペット機能を利用するには、以下のような順序で処理を行います。

1. パペット機能を有効にする。
2. パペット化したキャストを表示する。
3. パペット化したキャストに座標情報を与える

それでは実際のスクリプトを見てみましょう。EscapeG.LCP 中のスクリプトキャスト 4 番「プレイヤースコア_初期状態」を開いてください。3～5行目に

```
(* プレイヤーのパペット化 *)
TrackProperty [ プレイヤースコア:1 ].Puppet = True;
TrackProperty [ プレイヤースコア:1 ].Variable.Visible = True;
```

とあります。また、スクリプトキャスト 3 番「プレイヤースコア_立ち状態」の下部の方(41～43行目)には、

```
TrackProperty [ プレイヤースコア:1 ].Variable.Pos.X = 座標X;
TrackProperty [ プレイヤースコア:1 ].Variable.Pos.Z = 座標Z;
TrackProperty [ プレイヤースコア:1 ].Variable.Rol.Y = 回転値;
```

とあります。これを先程のパペット機能の使用手順と照らし合わせて、順に見ていきましょう。

3-3-1. パペット機能を有効にする

スクリプトでいうと⑥の1行目がこれに当たります。書式は、

```
TrackProperty[[Score:]Track].Puppet=true;
```

となっています。これは、

```
ScoreのTrack番号で示されるキャストをパペット化します。
```

という意味です。Score でスコア番号を、Track でトラック番号を指定します。また、Score は省略可能であり、その場合にはホレットスコアが選択されます。最後の true を false にすると既にパペット化されているキャストのパペット

ト化解除を行い、その後はスコアの情報に従うこととなります。

3-3-2. パペット化したキャストを表示する

スクリプトでいうと⑥の2行目がこれに当たります。書式は、

```
TrackProperty[[Score:]Track].Variable.Visible = True;
```

となっています。これは、

```
ScoreのTrack番号で示されるキャストを表示する
```

という意味です。ただし、この前に、そのキャストがパペット化されていなければなりません。最後の true を false にすると、パペット化されているキャストは表示されません。しかし、パペット化は解除されていないので、スコアの情報に従うというわけではありません。再度この Visible を true に設定すれば、同じ位置に表示されます。

3-3-3. パペット化したキャストに座標情報を与える

スクリプトでいうと⑦は全てこれに当てはまります。書式は、

```
TrackProperty[[Score:]Track].Variable.Pos.X = 代入情報;
```

となります。左辺最後の Pos.X の部分を色々変えて様々な座標情報を代入していくこととなります。

この Pos.X の部分は、「TrackProperty.Variable プロパティ」と呼ばれます。座標情報に関するプロパティの一覧を Table3 に示します。また、代入される情報は実数 (Float) 型となります。

DigitalLoca チュートリアル

Table3

プロパティ	説明
Pos.X	X座標を示します。
Pos.Y	Y座標を示します。
Pos.Z	Z座標を示します。
Rol.X	X軸方向の回転値を示します。
Rol.Y	Y軸方向の回転値を示します。
Rol.Z	Z軸方向の回転値を示します。
ScI.X	X軸方向の拡大率を示します。
ScI.Y	Y軸方向の拡大率を示します。
ScI.Z	Z軸方向の拡大率を示します。

※これはモデル、カメラ、ライトキャストの場合です。

このようにして代入された座標情報をもとに、パペット化されたキャストはその表示位置が決定されます。これらはスコア上においてキーフレームを設定するときの設定情報と全く同じです。これらのプロパティに数値を直接代入することにより、パペット化したキャラクタの表示位置を自在に設定できるのです。

3-4. 衝突判定

最もゲームらしい処理の1つと言えるのが衝突判定です。EscapeGuyでも、衝突判定を行っている箇所としては、

- 1.主人公の手とドアの衝突(主人公がドアを殴ったことを判定する為)
- 2.主人公の手と敵の衝突(主人公の攻撃結果を判定する為)
- 3.敵の攻撃[手・銃の弾・レーザー]と主人公の衝突(敵の攻撃結果を判定する為)
- 4.主人公と各種アイテムの衝突(アイテムの取得を判定する為)

といった所が挙げられます。2つの3Dオブジェクトの3次元空間内における位置関係を算出するわけですから、通常だと、多くの計算式を並べて衝突を判定するための処理行程を作らなければなりません。しかし、デジタルロケでは簡単に衝突を判定する為の命令が用意されています。EscapeGLCPの中のスクリプトキャスト61番「プレイヤーの攻撃が当たった時」を開いてください。7行目に、

```
if CollisionCheck(プレイヤースコア:2,ザコAスコア:3) then
```

とあります。ここでは、プレイヤーが攻撃を行ったときに、その攻撃が敵(ザコA:最初に遭遇する廊下にいる敵)に当たったかどうかを判定しています。この“CollisionCheck”が、簡単に衝突判定を行う為に用意された命令です。“CollisionCheck”は次のように宣言される関数です。

```
function CollisionCheck([ScoreA:]TrackA,[ScoreB:]TrackB) : Boolean;
```

これは、

```
ScoreAのTrackA番号で示されるキャストと、ScoreBのTrackB番号で示されるキャストが  
衝突しているか調べ、衝突しているようならtrueを、していないならfalseを返しなさい。
```

という命令です。2つのオブジェクトの衝突判定は、この1つの命令で行うことができます。

・章末

本章では、EscapeGuy で使用されている代表的な命令を学習してきました。次章からは、こういった命令群をどのように組み合わせて、コンテンツを作成するかを学習していきます。

第4章：コンテンツを作ろう

・はじめに

本章では、チュートリアルガイドで作成した「Walker」コンテンツのスキプトの全てを解説します。非常に小さなコンテンツですが、逆に、コンテンツにおけるスキプトの役目が分かりやすいとも言えます。また、コンテンツを作成する際は、スキプトだけでなく、スコアとの関係も非常に重要になってきます。意外と軽視されがちですが、スコアをどう作成するかによって、スキプトが全然違ってきます。そこで、本章ではスキプトだけではなく、随所でスコアに関する説明が入ってきます。スキプトを見たり、スコアを見たりと面倒になって行きますが、ゆっくり解説しますので落ち着いて学習していきましょう。

・サンプルコンテンツ

それでは、まず、本章で学習していく「Walker」コンテンツを見てみましょう。デジタルロケをインストールしたフォルダから「¥Tutorial¥Walker¥Completion¥ Completion.LCP」を開いて実行してください。先程作成した「Walker」コンテンツと比べると、若干修正が加えられていて、よりコンテンツらしくなっていますね。このコンテンツを本章では学習していきますから、まずはどんなコンテンツがじっくり試してください。操作に用いるのはカーソルキーのみです。

・ストーリー

コンテンツを作成していくとき、最初に必要な作業は何でしょう？モデリング、アニメーション、スクリプティング、どれでしょう？実はどれもありません。最初に必要なのは、「どんなコンテンツを作成するのか？」ということを決めていく作業です。当たり前かもしれませんが、実際はどうでしょうか。いきなり、モデリングを行ってアニメーションをつけて、それから「さあ、どんなコンテンツにしようかな？」と考えることも多いのではないのでしょうか。確かに大まかなイメージだけを持って、行き当たりばったりで作って、問題無くコンテンツが完成するかもしれません。しかし、最初にどんなコンテンツを作成するかをしっかりと考えておくと、後々の作業がスムーズ（スマート）に進みます。前置きが長くなってしまいましたが、以上のようなことを踏まえて、まず「Walker」コンテンツの流れや、ストーリーを整理してみましょう。

・コンテンツの流れ

文章で、「Walker」コンテンツを説明すると、

Walkerを操作して、扉から脱出しよう。ただし、扉を開けるには、ステージ内に落ちているカギが必要だ。

といったところでしょうか。一応、脱出に成功するとエンディングとなるアニメーションがあります。それでは、この流れをフローチャート（プログラミングで用いる流れ図）風に見てみましょう。スコアと対応させて説明していきますので、スコアも必ずご覧になってください。

Fig1 Walker コンテンツ流れ図

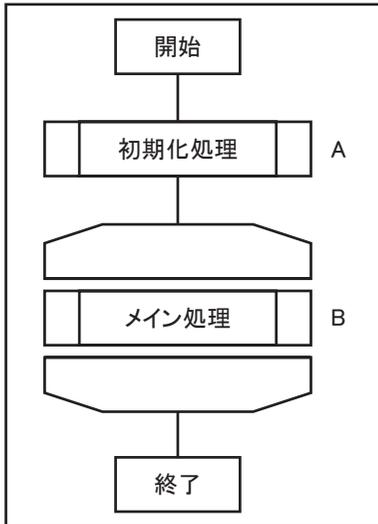


Fig1は最も大まかな流れです。それぞれの処理の中身及びスコアでの対応箇所は以下のようになっています。

- A.初期化処理…内容:コンテンツ内で必要な全ての初期化処理を行う。
 対応:ルートスコア0フレーム。
 B.メイン処理…内容:コンテンツのメインとなる処理。
 対応:ルートスコア1フレーム。

このように「Walker」コンテンツが再生されると、まず初期化処理が行われ、その後、実際にコンテンツのメインの部分（ゲーム部分）が処理され続けます。

・補足

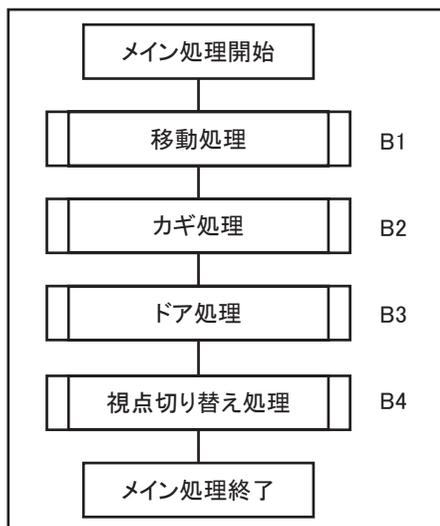
Fig1のメイン処理の上下を囲む記号は、繰り返し、つまりループを意味します。また、上側の記号内にはループの終了条件が書かれています。ですから、この例でいえば、「コンテンツ終了という判断がされるまで、メイン処理を繰り返しなさい。」ということになります。

あくまでも、これはフローチャート風の書き方であって、フローチャートではありません。今回、コンテンツの流れを説明しやすいように図化したものでしかありません。

初期化処理と言えば、ある程度は想像出来るかもしれませんが、メイン処理というだけでは漠然としすぎています。次のステップ、Fig2では、Fig1のメイン処理をもう少し詳細に見てみましょう。

DigitalLoca チュートリアル

Fig2Walker コンテンツ - メイン処理流れ図



このようにメイン処理では4つの処理を毎フレーム行っています。それぞれの処理は以下のようになっております。

- B1.移動処理…内容:主人公を移動させる処理。
対応:ルートスコア1フレーム。
- B2.カギ処理…内容:カギの状態をチェックする処理。
対応:カギスコア。
- B3.ドア処理…内容:ドアの状態をチェックする処理。
対応:ドアスコア。
- B4.視点切替…内容:視点を切り換える処理。
対応:カメラ切替スイッチスコア。

かなりコンテンツに近づいて来ましたね。それでは、こういったコンテンツの流れを踏まえた上で、スクリプトを読破していきましょう。個々のスクリプト単位で考えるのではなく、流れ図の処理ブロック単位で考えていきます。つまり、「A.初期化処理」、「B1.移動処理」、「B2.カギ処理」、「B3.ドア処理」、「B4.視点切替処理」の計5個のブロックでスクリプトを理解しましょう。

4-A. 初期化処理

4-A-1. 初期化とは

最初にコンテンツの初期化の部分を見てみましょう。ここでは、コンテンツを実行する際、最初に行わなければならない事が記述されています。今回の例では、

- | |
|---|
| 1.パペット化に関する手続き
2.変数等の初期化
3.BGM・背景等の設定 |
|---|

が行われています。1や3はコンテンツによっては不必要かもしれませんが、2に関しては、ほとんどのコンテンツで必要となる処理でしょう。以下に初期化部分のスクリプトを示します。これは、スクリプトキャスト0番「初期化」というキャストにあたります。

注：スクリプトの各行左端には行番号が書かれていますが、これは便宜上のものであり、実際には記入されません。

0 初期化

var	1行目
移動許可:	2行目
Boolean;	3行目
前回回転方向, 前回移動方向:	4行目
Float;	5行目
カギ取った:	6行目
Boolean;	7行目
OnEvent ExitFrame(Score,Track: Integer);	8行目
begin	9行目
(* カメラのパペット化 *)	10行目
TrackProperty [メインカメラスコア: 1]:Puppet = True;	11行目
TrackProperty [メインカメラスコア: 1]:Variable.Visible = True;	12行目
TrackProperty [ウォーカービューカメラスコア: 1]:Puppet = True;	13行目
(* ウォーカーのパペット処理 *)	14行目
TrackProperty [ルートスコア: 1]:Puppet = True;	15行目
TrackProperty [ルートスコア: 1]:Variable.Visible = True;	16行目
TrackProperty [ルートスコア: 1]:Variable.Pos.X = 230.0;	17行目
TrackProperty [ルートスコア: 1]:Variable.Pos.Z = -230.0;	18行目
TrackProperty [ルートスコア: 1]:Variable.Rol.Y = -PI / 2.0;	19行目
(* 変数の初期化 *)	20行目
前回回転方向 = 0.0;	21行目
前回移動方向 = 0.0;	22行目
移動許可 = True;	23行目
カギ取った = False;	24行目
(* BGM再生 *)	25行目
MidiAutoRepeat(True);	26行目
MidiOpen(バックミュージック);	27行目
MidiPlay;	28行目
(* バックグラウンドの指定 *)	29行目
WorldBG(星空);	30行目
end;	31行目
	32行目
	33行目

以上の処理内容より、全体を大きく区分すると、

DigitalLoca チュートリアル

1～4行:	グローバル変数の宣言
8～18行:	パペット化に関する手続き
20～24行:	変数等の初期化
26～33行:	BGMの再生、背景の設定

と区分出来ます。なお、空白行などは除きます。この内、「グローバル変数の宣言」及び「変数などの初期化」は、コンテンツで必要なグローバル変数を宣言し、初期値を代入しているだけです。ここでは、それぞれの変数がそのような意味を持っているかは無視することとして、説明を省略します。それでは残る2つ、「パペット化に関する手続き」、「BGMの再生、背景の設定」を見て行きましょう。

4-A-2. パペット化に関する手続き

このコンテンツでも、幾つかのキャストをパペット化して操作しています。パペット化されるキャストは以下の通りです。

1.メインカメラ	(メインカメラスコア1番トラック)
2.ウォーカービューカメラ	(ウォーカービューカメラスコア1番トラック)
3.ウォーカースコア	(ルートスコア1番トラック)

この内、カーソルキーの操作によって、座標値を変化させるのは3番のウォーカースコアのみです。あとの二つは表示状態を変化させるのみです。

メインカメラは斜め上方から、このコンテンツの世界を見下ろしている固定カメラです。もう一つのウォーカービューカメラは、ウォーカーの後部上方に固定され、常にウォーカーの後姿を写しているカメラです。これら二つのカメラは、コンテンツ再生スクリーン右上の二つのボタンによって切り換える事ができます。

※この部分に関する詳細は「B4. 視点切替処理」で解説しています。

この初期化スクリプトでは、上記3つのキャストのパペット化を行い(8～13行)、さらにウォーカーに関しては最初に配置する位置を決めています(15～17行)。

4-A-3.BGMの再生

初期化スクリプトの最後の部分では、BGMの再生と背景の設定を行っています。まず、BGMの再生です。デジタルロケではWAVデータとMIDIデータを音声データとして扱うことができます。この内、WAVデータは効果音を表現するのに使用します。コンテンツの再生中に流れるBGMには、もう一方のMIDIデータを使用します。今回の例では、

26行目MIDIが自動的にリピート再生されるように設定。
27行目BGMとするMIDIキャストを開く。
28行目MIDIを再生する。

という手順によって、BGMを再生しています。このコンテンツにおいてBGMに関する記述はこの3行のみです。

スクリプト入門

ここで注意しなければいけないのは、「MIDI を再生する前に、必ず該当する MIDI キャストを開いておく」ということです。つまり、28 行目の前に、必ず 27 行目の処理を行っておかなければいけないということです。それでは、各行に登場した新たな命令を見ていきましょう。

まず、27 行目の「MIDI が自動的にピート再生されるように設定」で使用されている MidiAutoRepeat 命令です。これは次の様に宣言されているものです。

```
procedure MidiAutoRepeat ( F : Boolean );
```

MidiAutoRepeat 命令は、MIDI キャストの再生が終了したときに、自動的に再生位置を先頭に戻し、再び再生を始めるかどうかを指定するものです。引数として True を与えればピート再生が設定されます。逆に False を与えると、再生は 1 回のみです。また、この設定は MIDI キャストが再生されていないときに行うようになっています。今回はピート再生に設定しています。

28 行目では MidiOpen 命令を用いて、MIDI ファイルを開いています。宣言は以下の様になっております。

```
procedure MidiOpen ( CastNum );
```

この手続きでは、MIDI キャストをオープンし、MIDI 再生の準備を行います。スクリプトリファレンスでは、「引数 CastNum には CastNumber 関数によって取得できる Integer 型の式、またはキャストの名称を直接指定できます」とありますが、ここでは MIDI キャストの名前を直接指定しています。今回 BGM として流す音楽は「バックミュージック」という名前の MIDI キャストです。

最後の 29 行目では、MidiOpen 手続きによって準備された MIDI キャストの再生を開始します。

```
procedure MidiPlay;
```

この命令は至極単純です。この 1 命令を書けば、開かれている MIDI キャストが再生されます。1 点だけ注意するのは、この命令の前に MIDI キャストを開いておくことです。

以上これだけで、BGM の再生ができます。BGM を流したいというだけなら、これで十分です。また、これら以外にも、デジタルロケには様々な MIDI 制御命令が用意されていますので、より細やかな処理を行うこともできます。スクリプトリファレンスを参考に、色々試してください。

4-A-4. 背景の設定

3D ワールドを構築する際、全てのものを 3D モデルで表すことが可能です。しかし、それには非常に多くのモデル、つまりポリゴンを必要とします。これでは座標計算が多くなり、コンテンツとして効率が良くありません。そこで、ビットマップキャストを後方に配置し、背景として用います。こうすることによって、画面の賑やかさを保ちつつ、ポリゴン数を減らすことができます。なんでも 3D にすれば良いというものでは無いのです。

ところで、上記のような方法で、ビットマップキャストを背景にすることができますが、この場合背景は固定されたままです。つまり、カメラアングルがどう変わろうと背景は全く動かないのです。それでも問題無い場合もありますが、多くの場合は違和感を感じてしまいます。そこで用意されているのが WorldBG 命令です。まずは、宣

DigitalLoca チュートリアル

言を見てみましょう。

```
procedure WorldBG ( CastNum );
```

この手続きでは、ステージの最も後方に表示され、カメラの向きに同期してスクロールするビットマップ画像を作成する事が出来ます。こうすると背景に指定されたビットマップキャストは左右に自然にスクロールするようになります。引数には、MidiOpen のときと同様、ビットマップキャストの名称、もしくは CastNumber 関数によって取得できる Integer 型の式を使用します。また、引数を -1 とすると、背景設定を解除することもできます。

以上、初期化処理ブロックに関して、どのようなことが行われているか見てきました。次からは、いよいよコンテンツのメイン処理の部分に入ります。

4-B1. 移動処理

ウォーカーコンテンツの最もインタラクティブなところといえば、カーソルキーに従ってキャラクター「ウォーカー」が歩き回るところです。この処理を行っているのが、これから説明する移動処理です。スクリプト的には、ルートスコアの 1 フレーム目に置かれている「移動処理」スクリプト (キャスト番号) に当たります。このスクリプトキャストの中で使用されている命令は、全て既に登場したもばかりです。ただ、組み合わせが少々複雑になっていますので、絡まった糸をほぐすように、少しずつ確実に理解していく必要があります。それでは、まず、そのスクリプトを見てみましょう。

1 移動処理

```
OnEvent ExitFrame(Score,Track: Integer);
var
  回転方向, 移動方向: Float;
  座標X, 座標Z, 回転値: Float;
begin

  (* 変数を初期化 *)
  移動方向 = 0.0; 回転方向 = 0.0;

  (* 移動不可能なら終了 *)
  if 移動許可 = False then exit; end;

  (* キーの読み込み *)
  if GetKeyState ( VK_UP ) then 移動方向 = -1.0; end;
  if GetKeyState ( VK_DOWN ) then 移動方向 = 1.0; end;
  if GetKeyState ( VK_LEFT ) then 回転方向 = -1.0; end;
  if GetKeyState ( VK_RIGHT ) then 回転方向 = 1.0; end;

  (* アニメーションの開始 *)
  if 前回移動方向 <> 移動方向 then
    if 移動方向 = -1.0 then SeekFrame ( ウォーカースコア : 前進 );
    else if 移動方向 = 1.0 then SeekFrame ( ウォーカースコア : 後退 );
    else if 回転方向 = 1.0 then SeekFrame ( ウォーカースコア : 右回転 );
```

```

else if 回転方向 = -1.0 then SeekFrame (ウォーカースコア: 左回転);
else
    SeekFrame (ウォーカースコア: 停止);
end;
end;
end;
end;
end;
else if (移動方向 = 0.0) and (前回回転方向 <> 回転方向) then
    if 回転方向 = 1.0 then SeekFrame (ウォーカースコア: 右回転);
    else if 回転方向 = -1.0 then SeekFrame (ウォーカースコア: 左回転);
    else
        SeekFrame (ウォーカースコア: 停止);
    end;
end;
end;
end;

end;

(* 変数にコピー *)
座標X = TrackProperty [ ルートスコア : 1 ]Variable.Pos.X;
座標Z = TrackProperty [ ルートスコア : 1 ]Variable.Pos.Z;
回転値 = TrackProperty [ ルートスコア : 1 ]Variable.Rol.Y;

(* 回転させる *)
:   回転値 = 回転値 + ( 回転方向 * 0.2 );

(* 向いている方向に進む *)
座標X = 座標X + ( 移動方向 * Sin ( 回転値 ) * 8.0 );
座標Z = 座標Z + ( 移動方向 * Cos ( 回転値 ) * 8.0 );

(* 移動範囲の制限 *)
if ( 座標X < 310.0 ) and ( 座標X > -310.0 ) then
    TrackProperty [ ルートスコア : 1 ]Variable.Pos.X = 座標X;
end;
if ( 座標Z < 310.0 ) and ( 座標Z > -310.0 ) then
    TrackProperty [ ルートスコア : 1 ]Variable.Pos.Z = 座標Z;
end;
TrackProperty [ ルートスコア : 1 ]Variable.Rol.Y = 回転値;

前回回転方向 = 回転方向; 前回移動方向 = 移動方向;
end;

```

これは、今回説明するスクリプトキャストの中で、最も記述の多いものです。最初に、どこで何をやっているか、大まかな流れを追ってみましょう。

DigitalLoca チュートリアル

2～4行:	ローカル変数の宣言
7～8行:	移動処理を行うかの判断
10～11行:	ローカル変数の初期化
13～17行:	キー情報の取得
19～37行:	アニメーションの決定
39～42行:	現在位置の取得
44～49行:	移動先の計算
51～58行:	キャラクタの移動
60行:	移動情報の保守

大きく分けるとこのような感じです。ここで、まず把握しておかないといけないのは、「位置の移動」と「アニメーションの変化」は個別に考えられているということです。私達が実際に歩くと、それは足を動かすという動作(アニメーション)によって、結果的に位置が移動することになります。しかしデジタルロケのコンテンツでは、この二つは別々に取り扱われます。歩く動作と位置の移動が繋がっては無いのです。ですから、作る方としては、移動に合わせてアニメーションを決めなければなりません。それでは、各処理を見てみましょう。

4-B1-1. ローカル変数の宣言

ここでは、位置座標やキー情報を蓄えておくために、必要なローカル変数を宣言しています。

4-B1-2. 移動処理を行うかの判断

市販のRPGゲームなどを思い浮かべてください。キャラクタを動かして何かイベントが発生すると、そのイベントが終了するまで、キー入力を受け付けなくなったりすることがよくあります。

ウォーカーコンテンツでは、クリア時に簡単なアニメーションが再生されます。そのときにユーザーがキャラクタを動かしてしまうと、アニメーションが成り立たなくなってしまいます。そこで、このコンテンツでも、クリア時のアニメーションが始まる前に、以後キー操作を受け付けられないような工夫がされています。

8行目を見てください。ここで「移動許可」という変数を見て、Falseの場合には「Exit」という処理をしています。この「Exit」を行うと、その後の処理が行われません。このコンテンツで、キー入力情報をチェックしているのは、このスクリプトだけです。逆に言えば、「移動許可」がTrueの場合にのみ、ウォーカーを操ることができます。

※初期化処理スクリプトの22行目で「移動許可 = True;」とされているので、初期状態ではウォーカーを操作することができます。

4-B1-3. ローカル変数の初期化

種々の計算で用いるローカル変数を初期化します。

4-B1-4. キー情報の取得

カーソルキーの入力チェックを行っています。その結果は、「移動方向」、「回転方向」というローカル変数に反映されます。この二つの変数の値は、Tableのような意味を持ちます。これがウォーカーの移動位置やアニメーションを決定するための情報となります。

Table 「移動方向」、「回転方向」の意味

変数名・値	-1.0	0.0	1.0
移動方向	前進	停止	後進
回転方向	左回転	進行方向維持	右回転

4-B1-5. アニメーションの決定

ここでは先程求めたローカル変数「移動方向」、「回転方向」の値と、さらに「前回回転方向」、「前回移動方向」というグローバル変数の値を基に、ウォーカーのアニメーションを決定しています。

「移動方向」、「回転方向」というローカル変数の値は、1フレームが終わると破棄されてしまいます。そこで、これらの値を「移動情報の保守」の部分でグローバル変数「前回回転方向」、「前回移動方向」に保存しています。これにより、常に1フレーム前のキー入力情報が分かる仕組みになっています。

ところで、なぜこのような処理を行うか疑問をお持ちになるかもしれません。例えば「移動方向」の値によってのみ、アニメーションを判定した場合で考えます。入力に応じて、スクリプト上では SeekFrame 命令によって該当アニメーションに移ります。この時、特定のカーソルキーを押し続けると「移動方向」の値は毎回同じ値をとります。そのため処理の度に、SeekFrame 命令が該当アニメーションの先頭フレームを毎回指定してしまいます。すると常に同じフレームが再生されてしまいます。これではアニメーションになりません。

そこで、「押したままである」という状態を把握するために、「前回回転方向」、「前回移動方向」というグローバル変数を使用して、1フレーム前の状態を残し比較するのです。

スクリプトではこれら4つの変数の値を比較・検討しながら、if 文によって、アニメーションを決定しています。if 文は「○○であって、さらに××であって、なおかつさらに…」と、入れ子構造のように記述することができます。非常に複雑に見えるかもしれませんが、1行ずつ丁寧に見ていくと、実は難しいものではないのです。

ここでは、全ての条件を列挙して、説明することは省略させていただきますが、基本的に以下のようなパターンでアニメーションを決定しています。

- 1.何も押されていないかったら、「停止」。
- 2.「回転」キーと「前進後退」キーが同時に押されている場合は、「前進後退」のアニメーションを再生。
- 3.同じキーが押されて続けている場合は、そのままアニメーションを流す。

以上のようなことは、アニメーションを伴うキャラクターの移動における基本パターンの一つです。必ず、1行ずつ理解してください。

4-B1-6. 現在位置の取得

新しい移動先を算出するために、まず、キャラクターの現在位置を取得して、計算用の変数に保存しています。

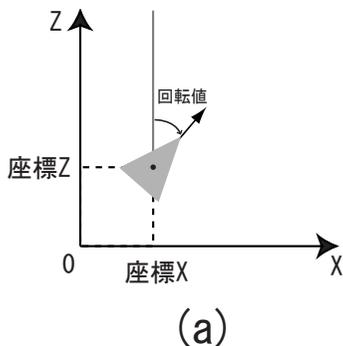
4-B1-7. 移動先の計算

このコンテンツでは、「1フレームの間に、どの方向にどれだけ動いた」という情報を計算し、その増加量を現

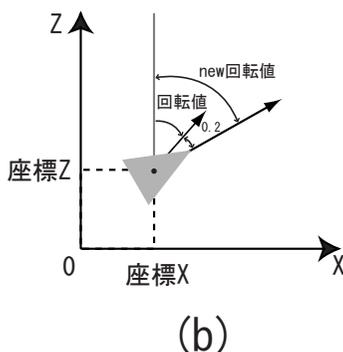
DigitalLoca チュートリアル

現在の座標に足してやることによって、新しい移動先の座標値を決定しています。

それでは、下の図を見ながら、移動先の計算方法を追っていきましょう。まずはスクリプトの 40～42 行目で取得した現在の状態を示します。



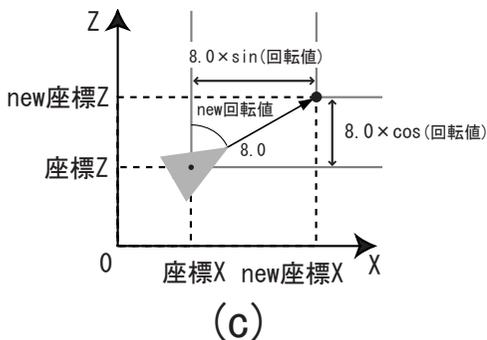
これが現在の位置です。次にスクリプトの 45 行目で以下の様に回転値に変化を加えます。



ここでは、キャラクターの新しい回転値を「回転方向」の値より決定します。左を向いた時は -0.2 が、右を向いたときは 0.2 が加算されます。今回は右に向いた例となっております。

※ 便宜上、新しい回転値に「new」を付けています。

次にスクリプト 48、49 行目で移動先の座標を決定します。



スクリプト 45 行目の計算で、進みたい方向が算出されたので、その方向に前進・後退させます。これには、新しい回転値を基に、移動量（ここでは 80）を X 軸成分・Z 軸成分に分解しています。

Sin(サイン)、Cos(コサイン) など懐かしいものが出てきましたが、図で示した程度の事です。難しい事ではありません。ここでは、上記のようなステップに沿って、新しい座標値を算出しています。

4-B1-8. キャラクタの移動

計算された移動先の座標値を、TrackProperty に代入することによって、キャラクターを移動します。その際、キャラクターが床からはみ出して歩いて行かないように、代入する座標値が床の枠に収まっているかチェックをかけています。ここでは、X 座標が -310.0 ~ 310.0 以内で、かつ Z 座標が -310.0 ~ 310.0 以内であることをチェックしています。その数値を越えた値は座標情報として更新されません。

4-B1-9. 移動情報の保守

「アニメーションの決定」で書いたように、ローカル変数「移動方向」、「回転方向」の値を、グローバル変数「前回回転方向」、「前回移動方向」に保存しています。このよう情報は、次のフレームで「移動処理」を行う際に利用されます。

4-B2. カギ処理

今回のコンテンツでキーとなるのは、文字通り「カギを取ったか」ということです。コンテンツを再生すると、まず初期化処理の部分で、「カギ取った」という Boolean 型の変数に False が代入されています。この「カギ取った」の値が False” ということで、カギをまだ持っていないという状態を示しています。それでは、「カギ取った」が False から True に変化するのはいどこでしょうか？この状態を監視しているのが次のスクリプトです。

2 カギを取った処理

OnEvent ExitFrame(Score,Track: Integer);	1行目
If CollisionCheck(カギスコア:4,ウォーカースコア:2) = True then	2行目
SeekFrame(カギスコア:なくなる);	3行目
SeekFrame(画面表示スコア:カギ表示);	4行目
カギ取った = True;	5行目
end;	6行目
end;	7行目

これは、カギスコアの 1 ~ 80 フレームに置かれているスクリプトです。つまり、コンテンツ内で 3D のカギが回転アニメーションをしている間、常に実行されるスクリプトです。

構造はいってシンプルで、if 文が一つあるだけです。この if 文において条件式となっているのが、2 行目の

```
CollisionCheck(カギスコア:4,ウォーカースコア:2) = True
```

DigitalLoca チュートリアル

という部分です。これは両方のスコアを見て確認して欲しいのですが、カギのモデルとウォーカーの体の部分が衝突しているかどうかを調べています。そして衝突が確認されたときだけ、以下のような処理が行われるのです。

```
SeekFrame(カギスコア:なくなる);  
SeekFrame(画面表示スコア:カギ表示);  
カギ取った = True;
```

まず、3行目では、カギスコアの再生位置を「なくなる」というラベル名の場所に移動しています。この位置に来ると、まず、カギを取った効果音 (WAV キャスト「カギ取った音」) が鳴ります。その後は、何も無いフレームがループで再生され続けます。すなわち、ここで3Dのカギは非表示となり、さらに、前述の「カギのモデルとウォーカーの体の衝突判定」も行われなくなるのです。

次に、4行目では、画面表示スコアの再生位置を「カギ表示」というラベル名の場所に移動しています。このスコアでは、ここまで「キーボードの↑↓←→で操作してください。」と書かれた操作説明用のテキストキャストを表示していました。これが「カギ表示」ラベルに移って、「カギ」というビットマップキャストが画面右下に表示されるようになります。このビットマップによって、「ウォーカーがカギを取得した」ということを視覚的に示しているのです。

そして、最後の5行目で、「カギ取った」という変数の値を False から True に変えることにより「カギを持っている」という状態を表しているのです。

この3行によって、コンテンツとしての目的が「カギを取る」から「扉から脱出する」に変更されるのです。

4-B3. ドア処理

このコンテンツは、ウォーカーがカギを取った後、扉を抜けて脱出することが目的です。ですから、扉に関する処理を行う「ドア処理」は、コンテンツのストーリーを制御する重要な部分といえます。このドア処理は、「ドアスコア」で行われているのですが、具体的には以下のことが行われます。

```
扉を開くかどうかの判断、及び開いた際の各処理(エンディングの再生)
```

それでは、順にスクリプトを見ていきましょう。

3ドアが開く処理

OnEvent ExitFrame(Score,Track: Integer);	1行目
If CollisionCheck(ドアスコア:1,ウォーカースコア:2) = True then	2行目
If カギ取った = True then	3行目
移動許可 = False;	4行目
TrackProperty [ルートスコア: 1].Puppet = False;	5行目
TrackProperty [メインカメラスコア: 1].Variable.Visible = True;	6行目
TrackProperty [ウォーカービューカメラスコア: 1].Variable.Visible = False;	7行目
SeekFrame(カメラ切替えスイッチスコア:表示なし);	8行目
SeekFrame(メインカメラスコア:自動アングル);	9行目
SeekFrame(ウォーカースコア:ドア中入る);	10行目
SeekFrame(ドアスコア:ドア開く);	11行目
SeekFrame(画面表示スコア:終了);	12行目
end;	13行目
end;	14行目
end;	15行目

4-B3-1. コンテンツクリアーの判定

まず、2行目では、ドアのモデルとウォーカーの体の衝突を調べることにより、ウォーカーがドアの付近に来ているのかどうかをチェックしています。

次に、3行目では「カギ取った」が True か False か、つまりカギを持っているかどうかを調べています。この2、3行目において、条件を、

カギを持ってドアのそばに立っているとき

のように絞り込んでいます。この条件を満たしているときだけ、残りの部分の処理が行われます。満たしていないときは、ここでは何も行われません。

4-B3-2. コンテンツクリアー時の処理

さて、「カギを持ってドアのそばに立っている」というのは、このコンテンツのクリア条件ですから、あとは、エンディングへと向かうだけです。それでは、条件を満たしているときに何がされるか、1行ずつ読んでいきましょう。

まず、4行目ではグローバル変数「移動許可」を False にしています。これは、これから始まるエンディングアニメーションに備える為です。なお、「移動許可」は「移動処理を行うかの判断」の項を参照してください。

次に、5～7行目にパペット関連の命令があります。5行目では [ルートスコア:1]、つまりウォーカーのパペットを解除しています。これにより、今後、ウォーカーはスコアに書かれているキーフレーム情報に従って、動くことになります。

DigitalLoca チュートリアル

6、7行目では、「メインカメラ」と「ウォーカービューカメラ」の2つのカメラにおいて、メインカメラだけ機能するように設定しています。これに関しては、後述の「視点切換処理」で詳しく解説します。

8～12行目には、5つのSeekFrame命令があります。これらの処理は全て、これから始まるエンディングアニメーションのために行われるものです。それぞれの役割は以下のようになっています。

8行目 [カメラ切換スイッチスコア: 表示なし] 画面右上の視点切換用のビットマップを非表示にする。
9行目 [メインカメラスコア: 自動アングル] エンディング用にカメラを移動させる。
10行目 [ウォーカースコア: ドア中入る] ウォーカーがドアへ入っていくアニメーションを再生。
11行目 [ドアスコア: ドア開く] ドアが開くアニメーションを再生。
12行目 [画面表示スコア: 終了] 最後に「おしまい」というテキストキャストを表示。

以上のように、ここでは、コンテンツの最後、エンディングに関わる全ての部分を制御しています。

4-B4. 視点切換処理

リアルタイム 3D コンテンツでよく用意されている機能として、視点切換があります。通常 2D のコンテンツの場合、視点を変えるならば、それに対応した絵を予め用意しておかなければいけません。その点、3D ならば、カメラの場所を変更するだけですから、2D の場合に比べて非常に簡単に機能を実現することができます。

このコンテンツには、通常 (ユーザーがキャラクターを動かせる間)、2つの視点が用意されています。

1つは、初期状態で使用されている固定カメラです。これは、その名の通り、この世界に固定されたカメラで、常に斜め上方から、この世界を映しています。スコアでは「メインカメラスコア」が相当します。

もう一つは、ウォーカーの後方についているカメラです。これは、ウォーカースコアの子スコアとして設定されているので、ウォーカーの動きに追従して移動し、常にウォーカーの後方から世界を映します。スコアでは「ウォーカービューカメラスコア」が相当します。

この二つのカメラの切換によって、視点を変更するわけですが、この切換を促す部分、つまりスイッチが必要となります。その機能を果たしているのが「カメラ切換スイッチスコア」です。

このスコアでは、2つのビットマップが画面右上に表示されています。このビットマップをクリックすることにより、CastClick イベントが発生し、カメラの切換が行われます。この処理を行っているのは、次の2つのスクリプトです。

4 カメラ変更 VtoM

```

OnEvent CastClick(Score,Track,Button,X,Y: Integer);
    SeekFrame(カメラ切替えスイッチスコア:メインカメラon);
    TrackProperty [ メインカメラスコア: 1 ].Variable.Visible = True;
    TrackProperty [ ウォーカービューカメラスコア: 1].Variable.Visible = False;
end;

```

5 カメラ変更 MtoV

```

OnEvent CastClick(Score,Track,Button,X,Y: Integer);
    SeekFrame(カメラ切替えスイッチスコア:ウォーカービューカメラon);
    TrackProperty [ メインカメラスコア: 1 ].Variable.Visible = False;
    TrackProperty [ ウォーカービューカメラスコア: 1].Variable.Visible = True;
end;

```

両者とも行われているのは、以下の3つの処理です。

1. 選択視点が分かるようにスイッチのビットマップキャストを変更。
2. 選択されたカメラを表示状態に変更。
3. もう一方のカメラを非表示状態に変更。

特に難しい記述は行っていません。逆に言えば、これだけで視点変更という機能を実現する事が出来るのです。

・まとめ

以上5個の処理ブロックに分けて、ウォーカーコンテンツを見てきました。簡単すぎましたか?それとも難しかったですか?感じ方は人それぞれだと思います。しかし、このウォーカーコンテンツは、コンテンツ作成の基本となる部分が多く含まれていますので、難しかったという人はもう一度読んで理解を深めてください。また、「今回の例は簡単過ぎだけど、なかなか次のステップに行けない」という人は、このウォーカーをベースに色々機能を足してみるのはどうでしょうか?そうしている内に、全くのオリジナル作品を作る力が付くでしょう。

コンテンツを作成していると、ある特定の作業部分を集中して学んでしまうことが多いのですが、全体の構成をしっかりと整理することが大切だということを覚えておいてください。

・最後に

スクリプト学習に際し、「スクリプトリファレンス」を最初のページから読み始めて挫折するという方の話を良く聞きます。しかし「スクリプトリファレンス」はあくまで辞書でしかありません。一番の教材はサンプルコンテンツで

DigitalLoca チュートリアル

。「サンプルコンテンツを遊んでみて、動きを確かめて、そのスクリプトを読んでもみる」という作業を繰り返すことが、スクリプトを習得するための一番の近道なのです。ただ、これはある程度プログラミングの知識を持っていることを前提としていることも確かです。この「チュートリアルガイド - スクリプト編」は、そんな方の為に作成致しました。いわばデジタルロケスクリプトへの入り口となるものです。これが理解できれば、サンプルプログラムの内部構造を理解しやすくなりますし、「スクリプトリファレンス」の使用方法も本来の辞書的な方法になっているでしょう。スクリプトを使いこなせるようになるまでには、幾つもの壁があるかもしれません。その中でも一番最初の、そして一番多くの人を跳ね返してしまう壁を克服するために、本書によってお手伝いが出来ればと思います。

ユーザーインターフェイス と基本操作



0からのスタート

1 DigitalLoca ツアー

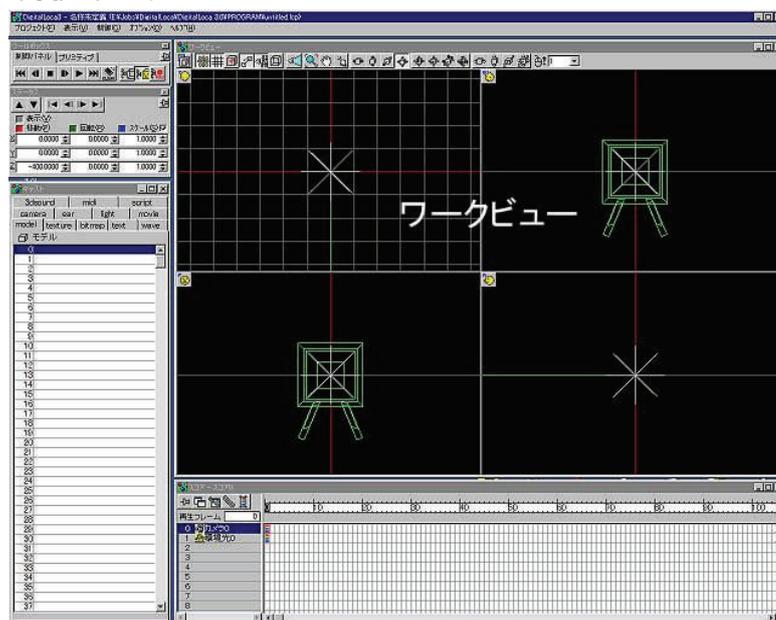
トピック

1) ワークビュー — カメラの配置

DigitalLocaを起動すると5つのウィンドウが表示されます。この中で右上にある1番大きなウィンドウが3Dモデルを3次元空間に配置する[ワークビュー]と呼ばれる作業領域です。

(FIG 1-1)

FIG 1-1



◆ バードビュー(鳥瞰図)



◆ カメラビュー



◆ 視点の前後移動

ショートカットキー
'Ctrl'+ 'Shift'+
マウスドラッグ



画面をDigitalLocaツアーの解説に合わせる為、一度メニューバーの[オプション/ウィンドウ位置の初期化]を実行してください。(FIG 1-2)

ワークビュー右上の顔のアイコンは、バードビューを表わしています。カメラからの視点にする為、アイコンをクリックして、カメラビューに変更します。(FIG 1-3)

次にメニューボタンの[視点の前後移動]ボタンを選択してから、マウスを下方向にドラッグ(ズームバック)して、各ビューの大きさを調整します。上面図(左上のビュー)上でグリッド全体が見渡せるぐらいまで調整してください。(FIG 1-4)

FIG 1-2



FIG 1-3

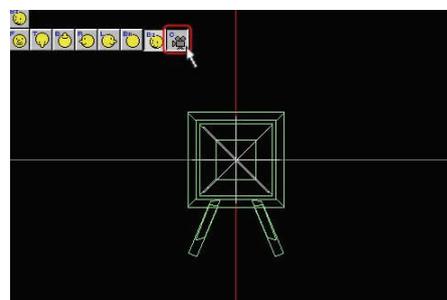
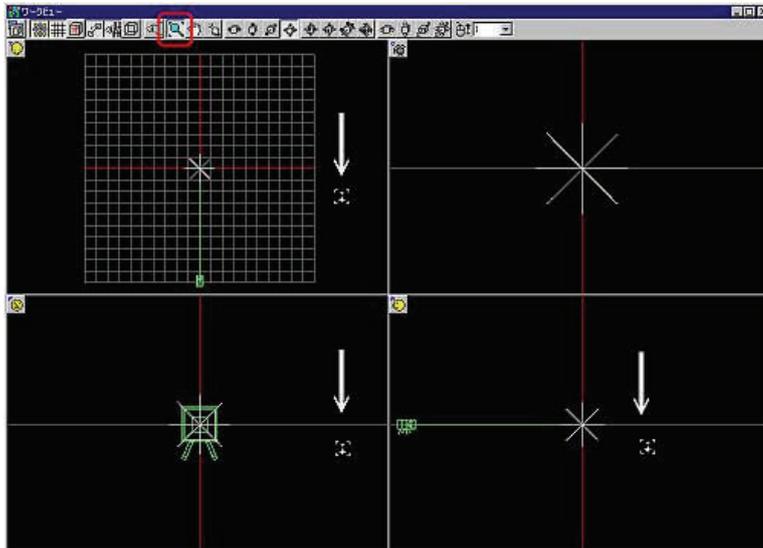
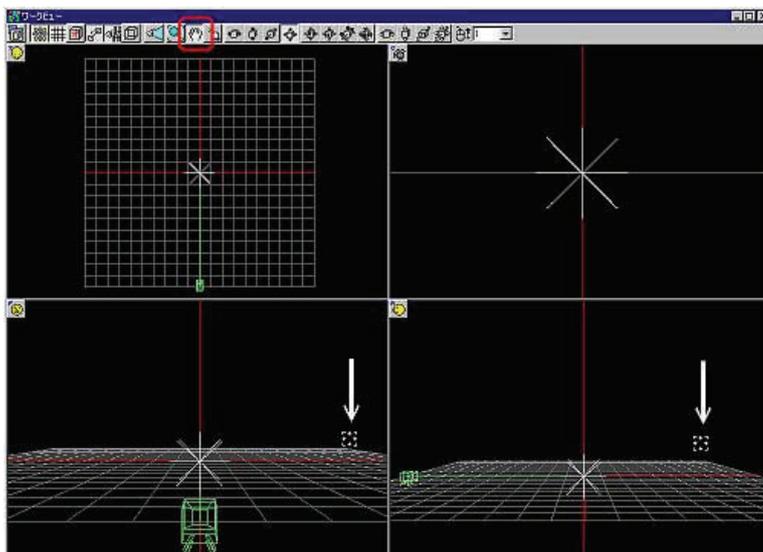


FIG 1-4



正面図(左下)と右側面図(右下)の2つビューは、メニューの[視点の上下/左右移動]ボタンを選択してから、マウスを下方方向にドラッグしてそれぞれ見やすい角度に調整します。なお、視点の移動調整をしてグリッドが斜め上方から見える表示になるのは、メニューにある[表示の切替え]ボタンが[パース表示]の場合です。[平行表示]になっている場合は再度ボタンを押して切り替えてください。(FIG 1-5)

FIG 1-5



初期状態では、カメラキャスト[カメラ0]が3次元空間の奥行きを表すZ軸座標上の[-400]に登録されていますが、高さを表すY軸上の高さが[0]であるため、カメラビューは、中心を見据えたままになっています。[カメラ0]の位置を変えるには画面の下にある

トピック

- ◆ 視点の上下/左右移動
'Shift'+'マウสดラッグ'



- ◆ 表示の切替えボタン

パース表示



平行表示



トピック

◆ 上下移動ボタン

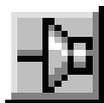
ショートカットキー
‘3’



MEMO)

ワークビュー各機能のショートカットは、ウィンドウがアクティブになっている時のみ有効です。

◆ ツールボックスのボタン



■ 前面表示固定ピン
常時前面に表示するボタン。



■ プリミティブ移動ボタン
矢印をクリックすることで他のプリミティブ機能が順次表示されます。

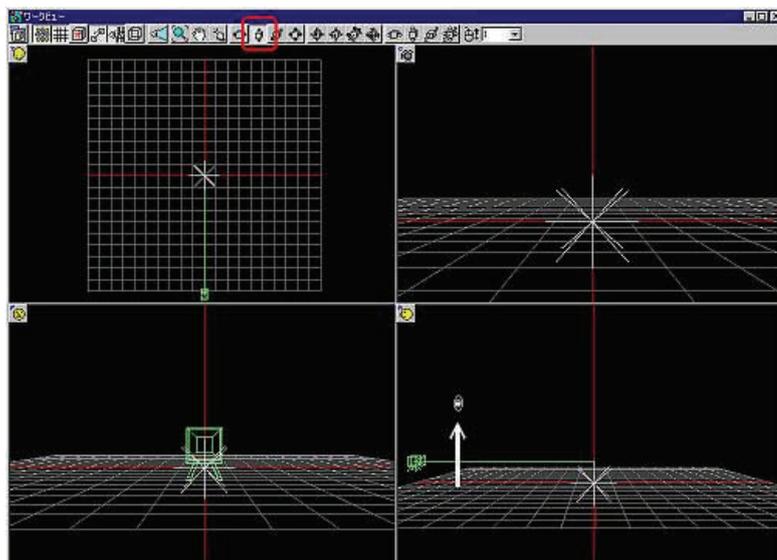
[スコア0]ウィンドウ内で[カメラ0]トラックのフレーム0を1回クリックして選択状態にしてから行います。
(FIG 1-6)

FIG 1-6



ワークビュー内で、[カメラ0]が緑色の選択状態になっていることを確認します。メニューの[上下移動]ボタンを押して、右側面図(右下)上で、マウスを上方向に任意にドラッグして高さを調節します。カメラからの映像はカメラビュー(右上)で確認することができます。(FIG 1-7)

FIG 1-7



2) ツールボックス機能とプリミティブキャスト

画面の1番左上にあるパネルが、再生制御を行ったり、基本=プリミティブキャストを簡単に作成できる[ツールボックス]です。

3次元空間には[カメラ0]と[環境光0]以外何も登録されていないので、ツールボックスの機能を使って簡単なキャストを作成してみましょう。まず、ツールボックスのタブを[制御パネル]から[プリミティブ]に切替え、1番左にある[立方

FIG 1-8



体]ボタンを押します。
(FIG 1-8)

[プリミティブの作成(立方体)]設定パネルが表示され、ここでサイズや分割数を決めることができますが、今回は初期設定のままで[OK]ボタンを押します。(FIG 1-9)

続いてプリミティブの左から2番目にある[平面]ボタンを押します。(FIG 1-10) [平面]は3Dモデルを置く地面にするので、以下の設定内容にします。

- モデルのサイズ設定
X[400] Y[400]
- [90度傾ける]にチェック

プリミティブ(タブ)内からは、上記の3Dモデルキャストだけでなく、ライトやカメラも追加することができます。現在ワークビュー内には、[環境光0]という最低限の基本光しかないのですが、他の種類のライトを追加することにします。1番右にある電球の形をした[ライト]ボタンを押します。(FIG 1-12)

[ライトの新規作成]パネルが表示されたら[平行光]を選択します。(FIG 1-13)

3) キャストウィンドウでできること

画面左下にある縦長のリストが[キャストウィンドウ]です。[ツールボックス]より作成されたプリミティブキャストは、自動的にこの

FIG 1-9

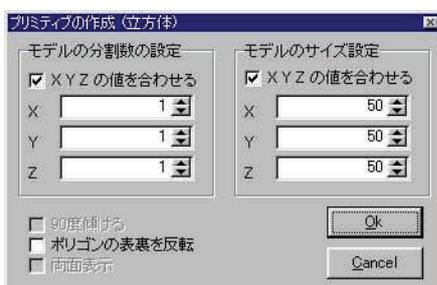


FIG 1-10

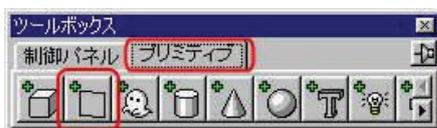


FIG 1-11

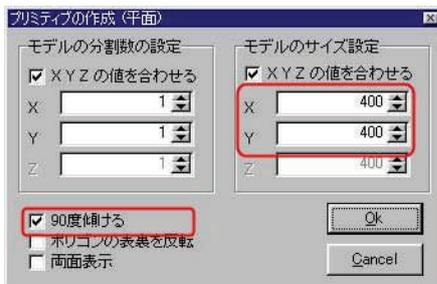


FIG 1-12



FIG 1-13



トピック

◆ プリミティブの設定項目

■ モデルの分割数設定
[XYZの値を合わせる]にチェックがある場合はXYZのの値が同じになります。最初に入力した値が他の座標値にも適用されます。

■ モデルのサイズ設定
[XYZの値を合わせる]にチェックがある場合はXY(Z)の値が同じになります。最初に入力した値が他の座標値にも適用されます。

■ [90度傾ける]
チェックがある場合は、XZ座標面上(グリッドと平行)に配置されます。

◆ 環境光

環境光は3Dモデルに最低限の均一な明るさを与える基本光です。環境光のみでは光の強弱といった3Dモデルの質感を表現することができません。環境光は、DigitalLoca起動時に自動的に登録されます。

◆ 平行光

平行光は、光の向き(角度)を持つライトです。角度を調整することはできませんが、光源の位置や強さをかえることはできません。現実の光源では太陽光の性質に相当します。

トピック

◆ キャストのナンバー

追加されたキャストは自動的に末尾ナンバーがつけられます。

◆ プリミティブキャストとキャストウィンドウ

ツールボックスのプリミティブから作成されたキャストはスコアとワークビューに自動的に登録されますが、キャストウィンドウ内で読み込まれる素材や新規作成されるmodel・camera・lightの各キャストはスコアにドラッグ&ドロップされるまでは登録されません。

◆ テクスチャの制限

DigitalLocaで扱えるテクスチャには以下の制限があります。

■ 種類

Windows Bitmap(bmp)
JPEG(*.jpg、*.jpeg)
Portable Network Graphics
(* .png)

■ サイズ

縦・横が2の累乗サイズ
例) 256×256
128×64 etc

ウィンドウの各キャストタブ内にリストアップされます。先ほど[平行光]を追加する作業をしたので、現在[light]タブ内のキャストが表示されているはずです。デフォルトで登録されている[環境光0]と、[平行光0]があることを確認してください。(FIG 1-14)

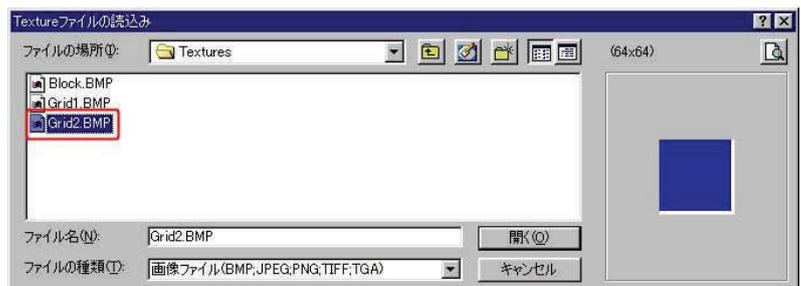
同様に、[model]タブをクリックして初めに作成した[立方体0]と[平面0]があることを確認します。(FIG 1-15)

外部ファイルから素材等を読み込む場合も、キャストウィンドウ上から行うことができます。では、3Dモデルに貼るテクスチャ(模様)素材を読み込んで見ましょう。

タブを[texture]に切替え、リストの空欄をダブルクリックしてください。(FIG 1-16)

[textureファイルの読み込み]ダイアログボックスが開かれたら、このDigitalLocaチュートリアルツアーの素材が入っている[Tutorial_Tour / Textures]フォルダを開き、[Grid2.bmp]という画像ファイルを選択します。(FIG 1-17)

FIG 1-17



テクスチャの適用方法については後で詳しく解説することにして、ここでは他の種類の素材をキャストウィンドウから読み込むことにし

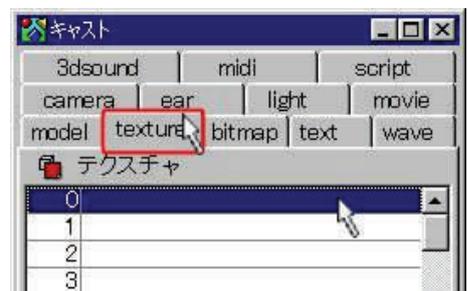
FIG 1-14



FIG 1-15



FIG 1-16

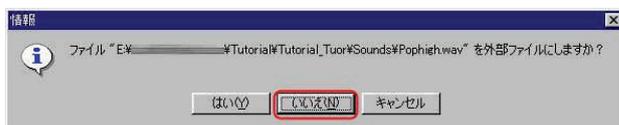


ます。キャストウィンドウのタブを [wave] に切り替え、リストの空欄をダブルクリックします。(FIG 1-18)

[効果音ファイルの読み込み]ダイアログボックスが開かれたら、[Tutorial_Tour / Sounds] フォルダを開き、[Pophigh.wav] という音声ファイルを選択します。(FIG 1-19)

音声や動画ファイルはサイズが大きくなる場合があります。その為、素材読み込み時のみ外部ファイルにするか、内部ファイルにするか問い合わせの [情報] が表示されますが、この Wave ファイルはサイズが小さいの [いいえ] を選択してください。(FIG 1-20)

FIG 1-20

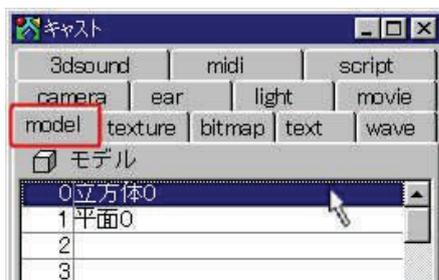


3) キャストのプロパティ

キャストウィンドウに登録された各キャストは、ダブルクリックすることで詳細設定や編集を行うプロパティウィンドウにアクセスすることができます。

では、モデルキャスト [立方体0] のプロパティウィンドウにアクセスしてみましょう。キャストウィンドウ [model] タブ内の [立方体0] をダブルクリックします。(FIG 1-21)

FIG 1-21



[モデル-立方体0] というプロパティウィンドウと、形状を確認する為の [モデルビュー] ウィンドウが同時に開きます。(次ページの FIG 1-22) 右側に表示される [モデルビュー] 内の形状は

トピック

◆ 外部ファイルとは？

HTMLコンテンツやマルチメディアCD-ROMと同様、DigitalLocaでもサウンドや動画データを本体ファイルと別に格納させることができます。これはサイズの大きな素材によるパフォーマンス低下を防ぐ役割があります。ただしコンテンツ全体のサイズが小さくなると予想される場合は内部ファイルにした方が良いでしょう。

トピック

◆ モデルビュー内の操作

■ 回転

‘マウドラッグ’

■ 表示の拡大・縮小

‘Shift’+‘Ctrl’+‘マウ
スドラッグ’

■ 表示のスライド移動

‘Shift’+‘マウドラッグ’

◆ 光源(ライト)と質感

3Dコンピュータグラフィックスでは、現実の空間と同様、物体の色は光源(ライト)に対する反射性質で決定されます。したがってモデル自身の色は[発光色]以外、光源の影響がないと表示することさえできません。DigitalLocaの初期起動時に最低限の光源である[環境光]があるのはその為です。

FIG 1-22



マウドラッグで自由に回転させることができます。(操作方法はトピック参照) 左側の大きなウィンドウでは、モデルの色・ハイライト色・発行色・透明度・テクスチャの割り当て等、質感を設定することができます。現在、[立方体0]の色はデフォルト色のグレー[R G B各値178]になっています。ウィンドウ内左側にある[質感設定/モデルの色]の四角いボタンを押して[色の選択]パネルを開いて変更してみましょう。

パネル内では好みの色合いや彩度をマウドラッグでピックアップすることができます。中央のカラーバーは明るさ、右のスライダーではそれぞれの色の要素を変更できます。入力欄に直接数値を入れることも可能です。ここではRG値をそのままにB値だけ[255]の明るいブルーにしてパネルを閉じることにします。

(FIG 1-23)

FIG 1-23



同様にして[ハイライト色]をRGB値[255.255.255]の白に設定します。[ハイライト指数]は、光源(ライト)に対する3Dモデル表面の反射度を決定します。スライダー右端の三角マークを1回だけクリックして値を[0.01]にします。今回は使用しませんが、この[質感設定]エリアでは3Dモデル自らが発色する[発光色]、透明度といった設定も可能です。続いてプロパティウィンドウの

左上にある▼マークを1回クリックしてキャストウィンドウ[model]タブのリスト2番目にある[平面0]の設定に移ります。(FIG 1-24)

FIG 1-24



プロパティウィンドウのタイトルバー表示が[モデル-平面0]に変わったことを確認します。[平面0]は地面にしたいので、わかりやすい柄の模様を貼り付けたいと思います。「テクスチャマッピング(柄模様を貼る)」と呼ばれるこの技法は3Dモデルにリアリティを与えることができます。まず、左側にある[テクスチャ1の設定]エリア内にある[テクスチャ]のプルダウンリスト▼マークをクリックして、リストの中にある[Grid2.bmp]を選択します。前にキャストウィンドウ[texture]タブ内で読み込まれた素材はこのリスト内に登録されるわけです。(FIG 1-25)

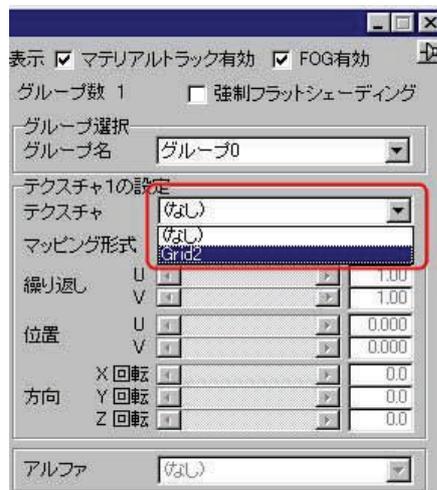
[モデルビュー]内の[平面0]を回転させて見てみるとテクスチャがうまく貼られていない感じがするはずですが、これはテクスチャマッピングの仕様として、3Dモデルに対して正面方向[Z軸方向]からマッピングされることが原因です。しかもテクスチャ[Grid2.bmp]は格子模様の必要最小限のパターン(トピック参照)内容である為、モデル全体に拡大されてマッピングされています。[テクスチャ1の設定]エリアの下側にある設定項目を以下の

トピック

- ◆ プロパティウィンドウ内でのキャストの移動とキャストウィンドウの関係

プロパティウィンドウとキャストウィンドウは互いに同期されているので、▲▼マーク、もしくはキャストウィンドウの対象キャストをクリックすることでも移動が可能です。

FIG 1-25



- ◆ [Grid2.bmp]の内容

格子(グリッド)パターンの最小単位



3Dコンピュータグラフィックス・システムの多くは、モデルに貼られるテクスチャの縦・横の位置や方向、繰り返し回数を任意に設定できる機能を持っています。

トピック

◆ テクスチャ方向(X回転)

テクスチャは基本設定では正面方向からマッピングされます。傾けられた「平面0」にマッピングする場合はX軸方向に90度回転させます。

◆ テクスチャとモデルの色

モデルの色の初期基本色はRGB各[178]のグレーなので、テクスチャの持つ色をメインにしたい場合は、モデルの色をRGB各[255]のホワイトにします。なお、モデルの基本色はメニュー／オプション／環境設定で変更することができます。

内容に調整して、地面に相応しい格子模様(グリッド)にしてみました。

(FIG 1-26)

[テクスチャ1の設定]エリア内

■ 繰り返し U方向(横)[10.00] V方向(縦)[10.00]

■ 方向 X回転[90.0] (*X軸を中心に90度傾ける為です。)

[質感設定]エリア内

■ モデルの色 RGB[255.255.255]

■ ハイライト色 RGB[255.255.255]

■ ハイライト指数 [0.01]

FIG 1-26

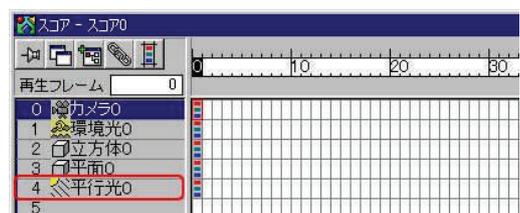


4) スコアの操作

最小限のモデルキャストを整える作業が終わりました。実際に3次元空間に配置するには、[スコア]と呼ばれる時間軸を持った表に登録する必要があります。[スコア]内にはアニメーションやスクリプトといった、演技や台本に相当する機能もあります。それでは画面下にある[スコア0]内に既に配置されている各キャスト=登場人物[立方体0]、大道具[平面0]、照明[平行光0]を操作して、3次元空間のステージを整えていきましょう。

主用キャストはすべてプリミティブツールから作成されているので、既にキーフレームがフレーム0に登録されています。トラック4にある[平行光0]を選択して、[環境光0]の下にド

FIG 1-27



ラッグして照明トラックどうしを揃えます。次にスコア中間にあるグレー枠のフレーム0をダブルクリックして繰り返しを設定する[ループ]マークを登録します。(FIG 1-28)

ツールボックスのタブを[制御パネル]にして、再生確認してみましょう。念の為左端にあるボタンを押して先頭フレームに戻し中央の再生ボタンを押します。(FIG 1-29)

再生画面は、[ステージウィンドウ]で表示されます。現在のスコア内容では[立方体0]位置や[カメラ0]の設定が正確でないことが確認できます。(FIG 1-30)

ステージウィンドウを閉じてキャストウィンドウ[camera]タブ内の[カメラ0]をダブルクリックします。(FIG 1-31)

カメラのプロパティである[カメラ-カメラ0]設定パネルが開きます。上にある[カメラの焦点距離]エリア内の設定を現在の[50mm]から[24mm]に変更してパネルを閉じます。(FIG 1-32)

FIG 1-28

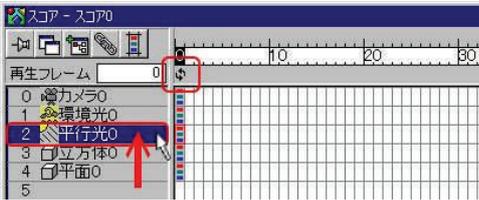


FIG 1-29



FIG 1-30

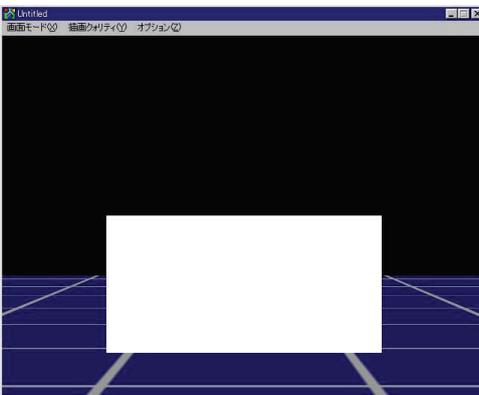


FIG 1-31

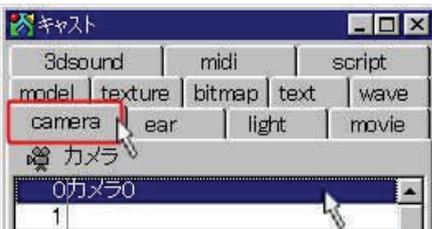


FIG 1-32



トピック

◆ ループ設定

スコア内では、時間軸に沿ってフレーム内の内容が移動再生されます。キーフレーム(各トラックのグレー表示の1コマ)が登録されていない場合でも移動再生が行われるので、ループ設定を登録して繰り返し再生させる必要がある。ループは一定フレーム区間に設定することも可能です。

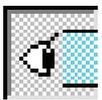
トピック

◆ 表示切替え

透視表示 (デフォルト)



平行表示



◆ 視点の前後移動

ショートカットキー
'Ctrl' + 'Shift' +
ドラッグ

◆ 視点のスライド移動

ショートカットキー
'Shift' + ドラッグ

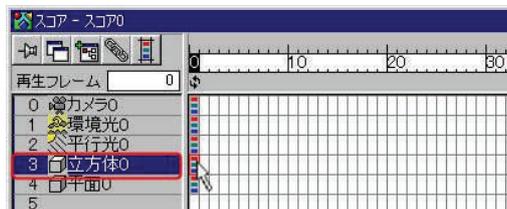
◆ 上下移動

ショートカットキー
'2'

再び「スコア0」に戻り、トラック[立方体0]のフレーム0をクリックして選択状態にします。

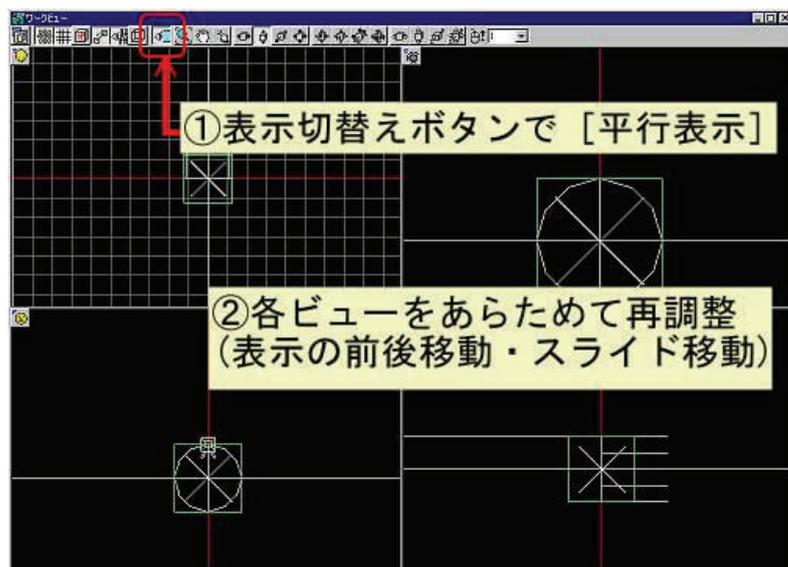
(FIG 1-33)

FIG 1-33



ワークビュー上で[立方体0]が選択状態である緑色表示になっていることを確認します。 [立方体0]を正確に[平面0]の上に移動させる為、上部メニューの左から8番目にある[表示切替え]ボタンを押して、現在のパース表示から平行表示に切替えます。 前に表示範囲を調整しましたが、表示方式によって独立した内容を保持しているので、あらためて再調整をしてください。 調整は[表示の前後移動]、[表示のスライド]ボタンを使用します。 なお、便利なショートカットキーが用意されているので覚えておくと便利です。(FIG 1-34)

FIG 1-34



ワークビュー画面左下にある正面図上に[立方体0]と[平面0]が表示されるように表示調整ができれば、上部メニューの[上下移動]ボタンを押して、正面図上で[立方体0]を上方向にドラッグします。 [立方体0]の底面と[平面0]のラインが重なるように移動調整してください。(次ページのFIG 1-35)

次に[スコア0]のトラック[平行光0]のフレーム0をクリックして選択状態にします。 現在[平行光0]は[平面0]と平行に正面に向いています。 3Dモデルの質感に変化を持たせる為、X・Y軸それぞれの角度を調整していきます。(FIG 1-36)

FIG 1-35

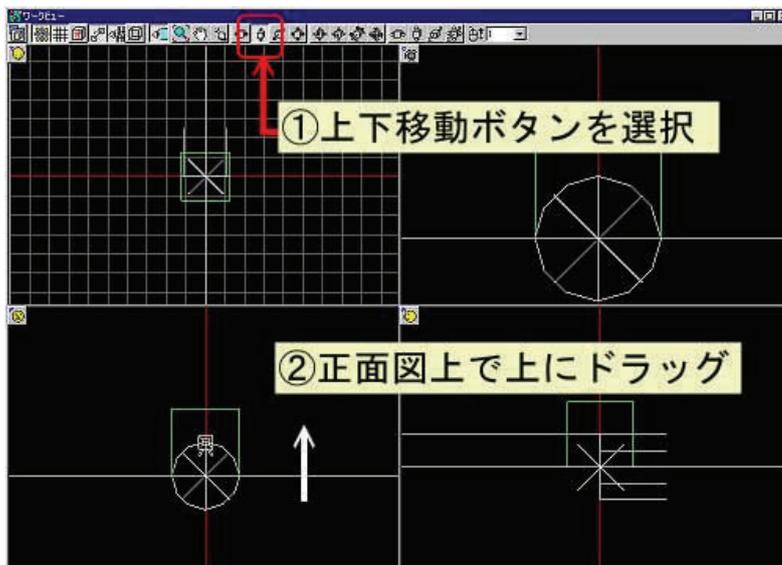
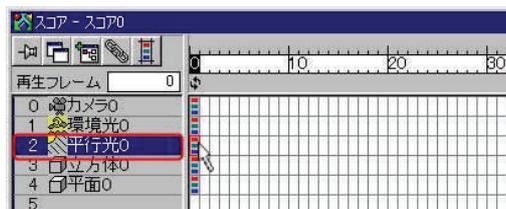


FIG 1-36



角度の調整は上部メニューの右から6番目にある[Z軸回転]ボタンを使用します。まず上面図上で下方向にマウสดラッグして約[-40]度ほど傾けます。次に側面図(右)上で上方向にマウสดラッグして約[40]度ほど傾けます。ワークビューの操作ボタンは、画面を操作する人の見た目のX・Y・Z軸を基本にしているので、[Z軸回転]機能1つで、上面図からの操作でY軸方向の回転、側面図からの操作でX軸方向の回転が可能になるわけです。(FIG 1-37)

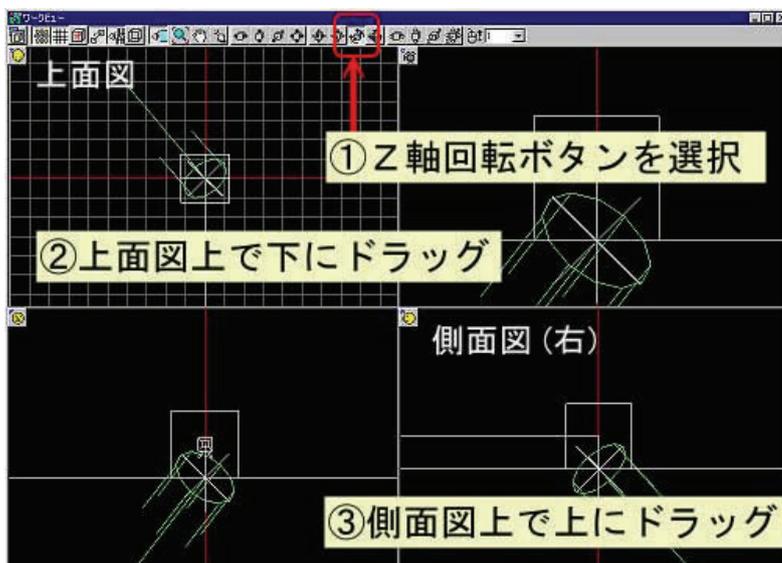


FIG 1-37

トピック

◆ Z軸回転

ショートカットキー
'7'



トピック

◆ 再生

ショートカットキー
‘Ctrl’ + ‘A’



◆ 再生画面を閉じる

ショートカットキー
‘Alt’ + ‘F4’

◆ コピー

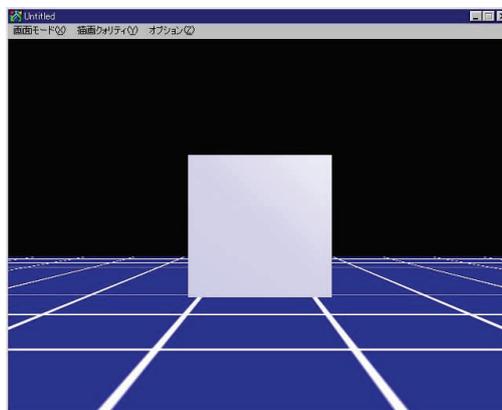
ショートカットキー
‘Ctrl’ + ‘C’

◆ 貼り付け(ペースト)

ショートカットキー
‘Ctrl’ + ‘V’

ツールボックス[制御パネル]タブの再生ボタンを押して再生確認します。[立方体0]が正面を向いて静止している為、画面的な変化はありませんが、初めの再生画面より質感に変化がでていることが確認できます。(FIG 1-38)

FIG 1-38



5) ステージウィンドウのリアルタイム再生

これまでの作業は、3Dリアルタイム再生を実現する為の下準備に相当します。したがって再生画面はまだ静止状態のままです。そこで、アニメーションとスクリプトを加えて簡単なインタラクティブコンテンツの体裁を仕上げていきます。

[スコア0]内の5つのトラック・フレーム0のキーフレームを上から下方向にマウスドラッグして全て選択します。マウス左ボタンを離すと選択状態になったフレームが赤枠で囲まれ、カーソルが手の形になります。そのままの状態でもマウス右クリックするとプルダウンメニューが表示されます。その中から[コピー]を選択します。

(FIG 1-39)

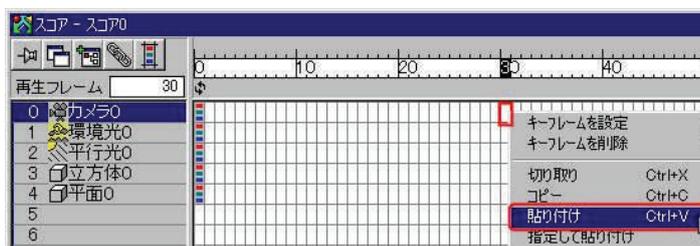
次にトラック[カメラ0]のフレーム30を1回クリックして選択状態にします。マウス右クリックによるプルダウンメニューの中から[貼り付け]を選択します。

(FIG 1-40)

FIG 1-39



FIG 1-40

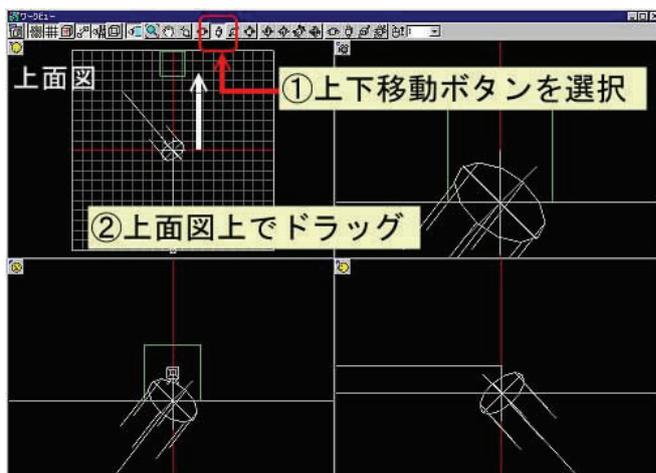


任意の空きフレーム(フレーム 30 以外)のクリックすることで貼り付けが確定され、同時に中間フレームが[表示]キーフレームを表わすグレーになります。(FIG 1-41)

次にトラック[立方体0]のフレーム15を1回クリックして選択状態にします。(FIG 1-42)

ワークビュー上で[立方体0]が選択状態である緑色になっていることを確認します。上部メニューの中から[上下移動]ボタンを選択してから、上面図上で上方向に[平面0]の端までマウスドラッグで移動します。(FIG 1-43)

FIG 1-43



トラック[立方体0]のフレーム15には[移動]キーフレームを表わす赤いマークが付きます。次にスコア上部のグレー枠部分に[ラベル]という標識名のようなものを設定します。フレーム0の上部をダブルクリックすると自動的にラベル名[▼L0]が作成されます。(FIG 1-44)

続いて隣のフレーム1もダブルクリックしてラベル名[▼L1]を作成します。フレームが接近しているのでラベル名[▼L0]が隠れ

FIG 1-41

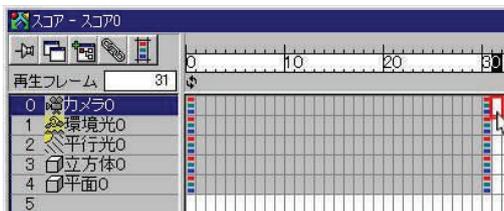
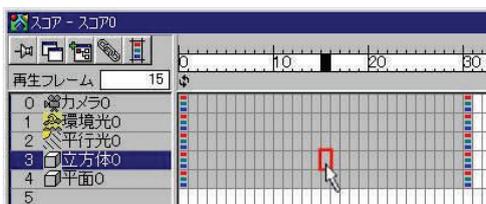


FIG 1-42



トピック

◆ 表示キーフレーム

3次元空間内に表示される時間を持つ(=存在する)意味を表します。スコア内ではコピー&ペースト(貼り付け)されたキーフレームは、中間フレームが自動的に[表示]キーフレームに登録されます。なお、[メニュー/オプション/環境設定]で、この自動機能をOFFにすることもできます。

◆ 移動キーフレーム

キーフレームは[表示]以外に、

- [移動] (赤)
- [回転] (緑)
- [スケール] (青)

の3種の要素があり、それぞれ単独に設定することができます。ただし、両端のキーフレームには3種の要素がすべて必要です。通常[表示]キーフレームを選択後、ワークビュー上での操作内容によって自動的に設定されます。スコア内で事前に設定したい場合は、マウス右クリックにより表示されるプルダウンメニューの中から選択します。前もってキーフレームを確定したい時等に便利です。

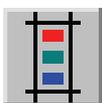
トピック

◆ ラベルの操作と編集

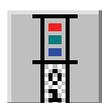
- 作成 ダブルクリック
- 移動 マウสดラッグ
- 名前変更
[▼]をクリック
- 削除 [▼]をダブルクリック

◆ スコア表示モードの変更

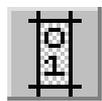
■ スコアモード



■ スコアとスクリプト



■ スクリプトモード



◆ スクリプト

スクリプト言語で記述されたプログラムのことをスクリプトとよびます。多くは処理手順をテキスト形式で記述します。

NOTE)

行末につける[;]セミコロンの入カミスに注意!

◆ CLEAR (初期化)

ショートカットキー
'Ctrl' + 'E'



てしまいますが、ラベル機能に影響はありません。

(FIG 1-45)

[スコア0]の左上にあるメニューボタンの右端にある[表示モードの切替え]ボタンを2回押してスクリプト表示モードに変更します。トラック[立方体0]のフレーム0をダブルクリックして[スクリプト]ウィンドウを起動させます。(FIG 1-46)

スクリプト表示モード時に、キャストが登録されたフレームをダブルクリックすると、キャストをクリックしたらイベントが発生する[OnEvent CastClick ~]というスクリプト定型文が自動作成されます。後は中間の行に命令文を挿入するだけで完成します。ここでは、

SeekFrame(L1);

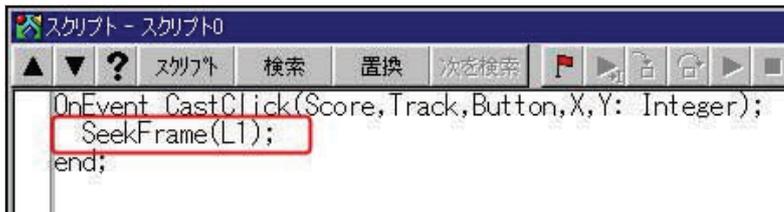
「[立方体0]がクリックされたらラベル名[L1]の付いたフレームを探しに行く(=SeekFrame)」

というスクリプトを追加することにします。(FIG 1-47・48)

FIG 1-47



FIG 1-48



ツールボックス[制御パネル]タブの[CLEAR 初期化]ボタンを押してから、再生します。(FIG 1-49)

FIG 1-49



FIG 1-45

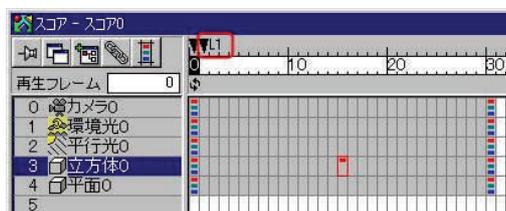
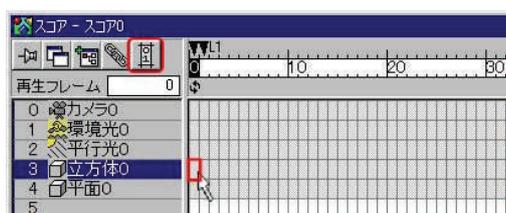


FIG 1-46



画面の立方体をクリックすると後方に移動するアニメーションが開始されますが、再度同じイベントが実行されないことがわかります。

これは再生ヘッドがアニメーションが終了した後も先に進んでしまうのが原因です。そこでアニメーションの終了する最後のフレームに、先頭に戻るスクリプトを設定することになります。このスクリプトは、キャストのあるトラックではなく、スクリプト専用のトラックに作成します。

一旦[スコア0]の表示を切替え、元のスコア表示に戻しておきます。次に任意のトラック(どれでも)をマウス右クリックして、表示されるプルダウンメニューの中から[ベーストラックの新規作成/スクリプト]を選択します。

(FIG 1-50)

作成されたトラック[スクリプト]のフレーム29をダブルクリックして、スクリプト

ウィンドウを起動させます。(FIG 1-51)

スクリプト専用トラックには、再生ヘッドがフレームを出たらイベントを発生する[OnEvent ExitFrame~]というスクリプト定型文が自動作成されます。(FIG 1-52)

FIG 1-52



ここに次の命令文を挿入します。(次ページのFIG 1-53)

SeekFrame(L0);

「再生ヘッドが(この)フレームを出たなら(=ExitFrame)ラベル名[L0]の付いたフレームを探しに行く

トピック

◆ スクリプトトラック

[ベーストラックの新規作成]によって作られるスクリプト専用トラックのフレームには、キャストナンバーの違うスクリプトをドラッグ&ドロップで複数登録することができます。ただし単純にフレームをダブルクリックした場合のみ、[OnEvent ExitFrame~]という定型文が作成されます。

なお、キャストウィンドウウィンドウに登録されたスクリプトをスコアの空きトラックにドラッグ&ドロップで新規登録する場合は、そのスクリプト1種類しか使用できません。

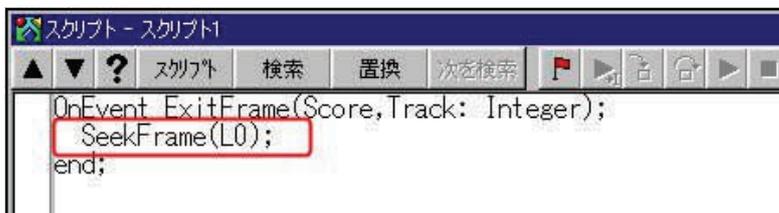
トピック

◆ Z軸回転

ショートカットキー
'7'

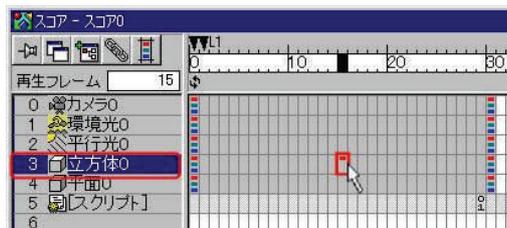


FIG 1-53



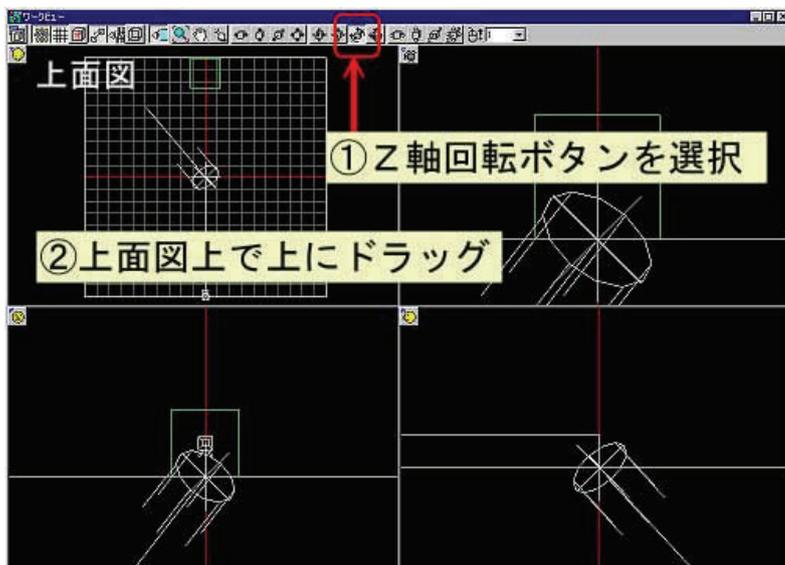
スクリプトの整合性が完成しましたが、[立方体0]のアニメーションは[移動]だけなので単調すぎます。そこで再びトラック[立方体0]のフレーム15を選択状態にして[回転]のキーを追加しようと思います。(FIG 1-54)

FIG 1-54



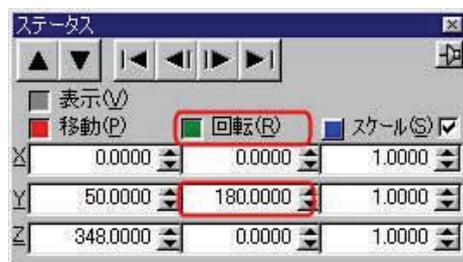
ワークビュー上で[立方体0]が選択状態である緑色になっていることを確認します。上部メニューの中から[Z軸回転]ボタンを選択してから、上面図上で上方向にマウสดラッグして[180]度回転させます。(任意の角度でも可) (FIG 1-55)

FIG 1-55



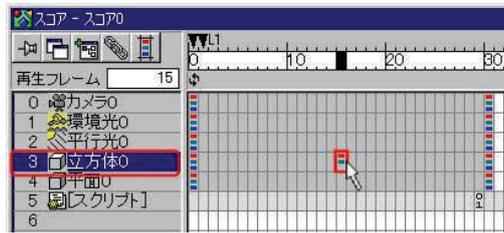
[移動]・[回転]・[スケール]のキーフレーム数値はツールボックスの下にある[ステータス]パネルに表示されます。スコア内でフレームを選択した後、

FIG 1-56



この[ステータス]パネル内の入力欄に直接入力してキーフレームを確定することも可能です。
(前ページFIG 1-56)

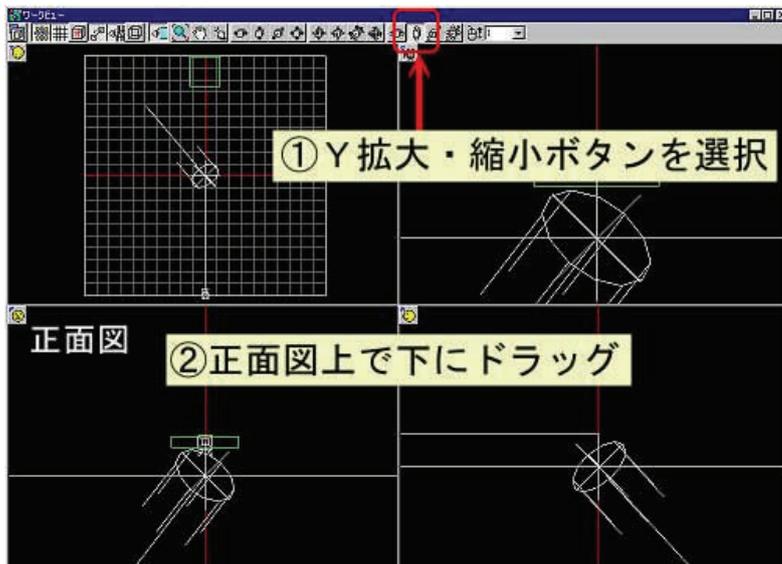
FIG 1-57



[スコア0]の方では、[立方体0]のフレーム15に[回転]キーを示す緑色のマークが付いています。(FIG 1-57)

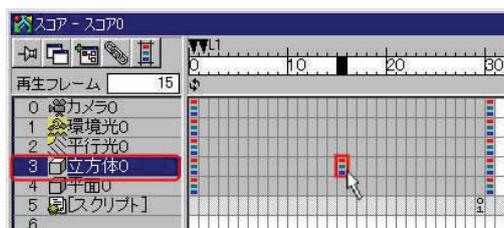
キーフレームはアニメーションの目的を考えて必要な要素のみを選択すべきですが、ここでは学習の意味を考えて残りの[スケール]キーも設定してみましょう。フレーム15が選択状態であることを確認して再びワークビュー上の作業に移ります。上部メニューの中から右から3番目にある[Y 拡大・縮小]ボタンを選択します。左下にある正面図上で上方向にドラッグして縦のスケールを縮小します。(FIG 1-58)

FIG 1-58



[回転]キーの時と同様、トラック[立方体0]のフレーム15に[スケール]キーを示す青色のマークが付いていることを確認します。(FIG 1-59)

FIG 1-59



トピック

◆ Y 拡大・縮小

ショートカットキー
'0'



◆ モデルの中心位置

FIG 1-58では、正面図上の[立方体0]がY軸スケールの縮小で[平面0]から離れて宙に浮かんだ状態になっています。これはモデル[立方体0]が座標の中心位置を基準に作成されている為です。[回転]・[拡大・縮小]の中心位置を変えたい場合は、モデルの編集で変更することができます。

トピック

◆ 拡張WAVEトラック

拡張されたWAVEトラックを使用すると、キーフレームは、WAVE制御用の情報を持つようになります。サウンドのオープン・クローズ、再生の開始・停止、繰り返し、音量や音質の変化(ドップラー効果)等、他のキャストのキーフレーム動作に合わせた制御が可能になります。(詳しくはオンラインユーザーズガイド/WAVEキャストを参照)

◆ キャストウィンドウから新規に登録されたキャストトラックのキーフレーム

トラックに新規登録されたキャストは、キーフレームを登録しない限り動作はしません。キーフレーム登録方法はキャストの種類によって異なります。

- ・[model]・[camera]
- ・[light]キャスト
フレームをダブルクリック、マウス右クリックによるプルダウンメニューからの選択で登録。
- ・[wave]・[bitmap]
- ・[script][midi]キャスト
フレームをダブルクリック

キャストウィンドウで作成されたキャストや読み込まれた素材は、ドラッグ&ドロップでスコアに登録します。[立方体0]をクリックした時に効果音が鳴るように、waveキャストをスコアに登録してみましょう。キャストウィンドウ[wave]タブ内の[Pophigh]を[スコア0]の空いているトラックにドラッグ&ドロップします。操作しにくい場合は、2つのウィンドウを並べて行ってください。

(FIG 1-60)

FIG 1-60



ドラッグ&ドロップ直後に「拡張されたWaveトラックを使用しますか」という情報メッセージが表示されますが、ここでは「いいえ」を選択してください。

(FIG 1-61)

FIG 1-61

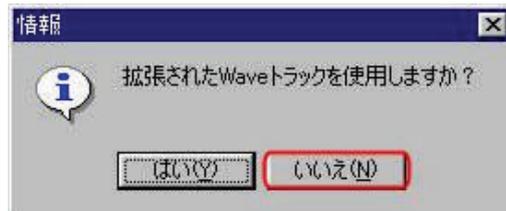


FIG 1-62

トラック[Pophigh]が登録されたら、アニメーションが開始するフレーム1をダブルクリックしてキーフレームを登録します。

(FIG 1-62)

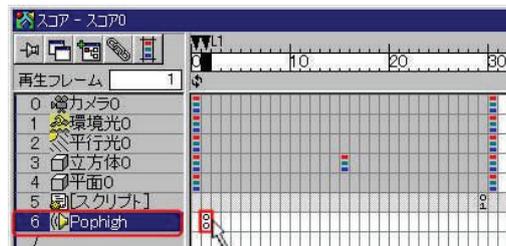
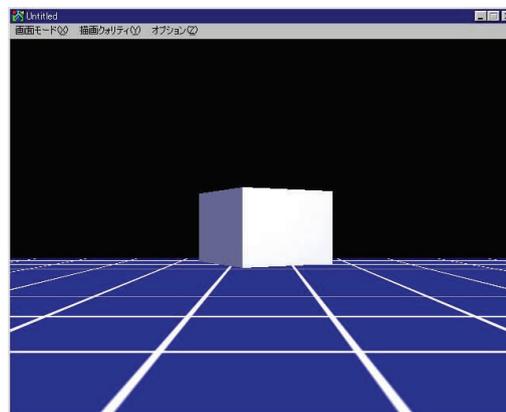


FIG 1-63

ツールボックス[制御パネル]タブの[初期化]ボタンを押してから、再生してみましょう。[立方体0]をクリックして音が鳴り、形状が変化するアニメーションを確認してください。

(FIG 1-63)



これまで何回となくリアルタイム再生画面を確認してきましたが、画面内のキャストがシンプルなものだけに、再生品質がちょっと気になります。特に[平面0]にマッピングされたテクスチャ表現に荒さが目立っています。DigitalLocaでは、リアルタイム3Dの再生品質を自由に制御する機能があるので、試してみましょう。

メニューバーの[プロジェクト]の中から[プロジェクトの設定]を選択します。(FIG 1-64)

[プロジェクトの設定]パネルが開いたら、[描画情報]というタブをクリックします。既に付いている項目はそのままにして、新たに以下の項目にチェックします。

[フォグ効果を有効にする]
[ミップマップ]

設定後、[OK]ボタンを押します。(FIG 1-65)

FIG 1-64

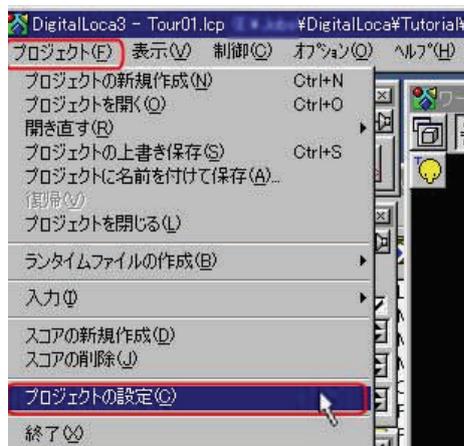
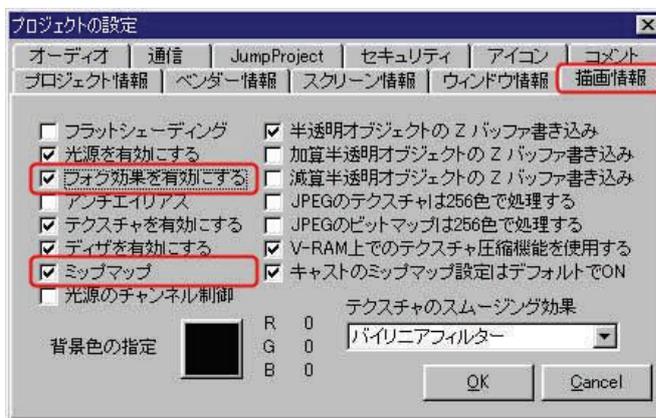


FIG 1-65



次にキャストウィンドウ [camera]タブ内の[カメラ0]をダブルクリックして、プロパティウィンドウを開きます。(FIG 1-66)

下段にある[フォグ設定]エリア内の[フォグ効果]にチェックを入れ、一番下にある入力欄の数値に次の値を

FIG 1-66



トピック

◆ ミップマップ

3Dオブジェクトにマッピングされるテクスチャのジャギー(ギザギザ)やモアレ(縞模様)を軽減する手法です。

1つのテクスチャに対し、複数の解像度のものを生成し、遠くにある部分には低い解像度のものを用い、近くにある部分には高い解像度のものを用います。通常1つの解像度のテクスチャから計算により自動生成されます。

トピック

◆ 「フォグの設定」

フォグ(霧)はカメラの視点方向からの距離情報をもとに徐々にかすむ効果です。フォグは最終描画面面に計算で生成されます。フォグ設定項目のうち、「前」はフォグのかかり始めの距離を意味し、「後」はフォグが完全にかかる距離を意味します。

このチュートリアルで設定する数値

- 前[400]
- 後[1200]

の値の根拠は、現在の「カメラ0」のZ軸方向の距離「-400」と「平面0」のサイズ「400×400」をもとに算出されています。つまり、[立方体0]の位置するあたりから徐々にフォグをかからせる為、カメラとの距離「400」から開始しています。完全にかかる[後]の[1200]は、カメラとの距離「400」プラス「平面0」大きさ[400]に、焦点距離の[24mm]の奥行きを考慮して決めています。[後]の数値は、後方に3Dモデルがないと曖昧になります。再生しながら数値を確認してみてください。

入力して修正します。

- 前[500]
- 後[1200]

(FIG 1-67)

[初期化]ボタンを押してから再生してみましょう。地面のグリッドが滑らかなり、フォグによる画面の奥行き感が増した表現ことが確認できるはずです。(FIG 1-68)

FIG 1-67

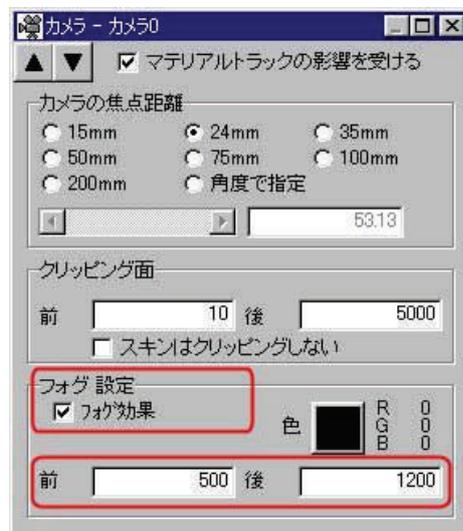
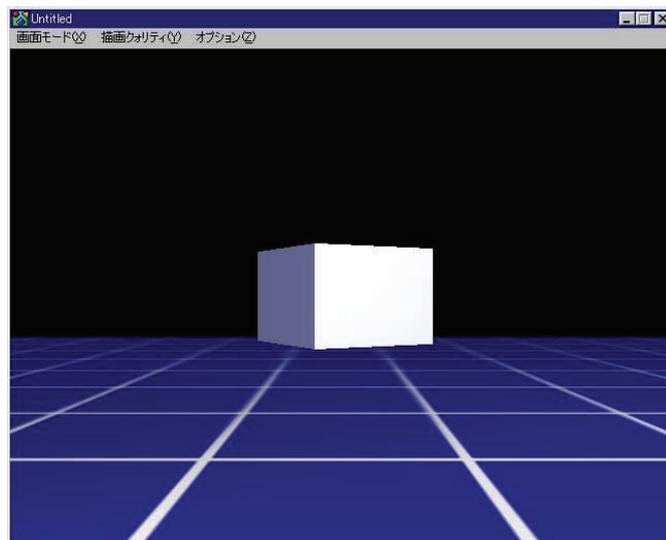


FIG 1-68



フォグの色はカメラのプロパティウィンドウ内で、自由に変更することができます。3Dモデルやカメラの位置、ライトの角度、カメラの焦点距離、フォグの設定をいろいろ変えてみると思わぬ効果が発見できるかもしれません。

最後にこのチュートリアル・プロジェクトを保存しておきましょう。保存はメニューバーの[プロジェクト]/[プロジェクトの保存]を選択して行ないます。

ポリゴンベースのモデリング



0からのモデリング テクニック

1 オフィスチェアへのモデリング

トピック

● トピックの使い方

トピックには、本文にでてくる用語の意味や操作、便利なショートカットキーが解説されています。合わせてご利用ください。

◆ 3面図

実際にはバードビュー(鳥瞰図)を含めて4面図になりますが、このビューは形状を確認するプレビュー的な意味合いがあります。したがって設計やCADの世界では上面図・正面図・側面図の3面図という呼び方が一般的です。

◆ 形状の編集



◆ バードビュー



◆ バードビューの塗りつぶし

ショートカットキー
'Shift' + 'F'

◆ バードビューのターン

ショートカットキー
'T'



(解除は右クリック)

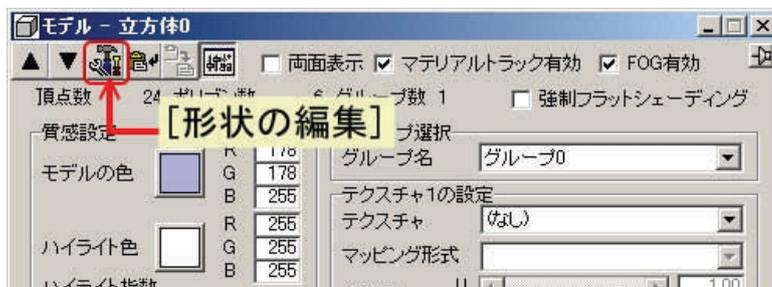
1) 形状を編集する

[ツールボックス]の[プリミティブ]に用意された基本形状は、コンテンツの目的にあった形に編集する必要があります。一般的にモデリングと呼ばれるこの編集作業では、モデルキャストの頂点や面を移動・追加することで複雑な形状に加工することができます。

キャストウィンドウ [model] タブ内の[立方体0]をダブルクリックして、[モデル-立方体0]のプロパティウィンドウを開きます。左上のボタンメニューの中から[形状の編集]を選択します。

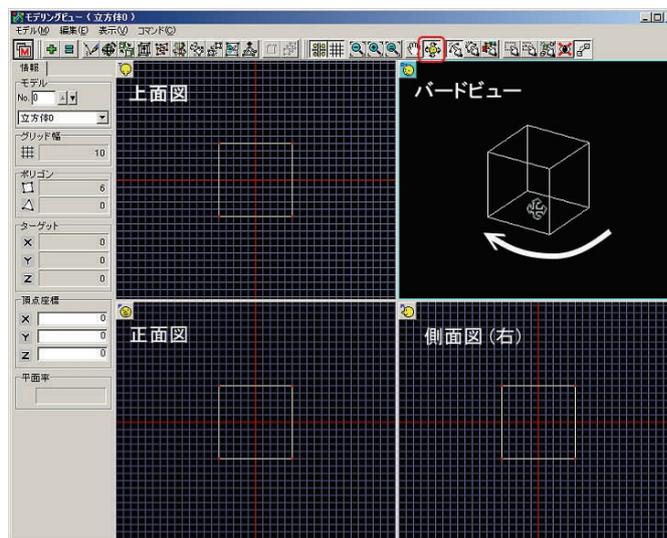
(FIG 2-1)

FIG 2-1



ワークビューと似たような3面図の[モデリングビュー]が起動します。各ビューの左上にあるアイコンは各ビューの向きを表しています。右上のビューは、編集の他に任意の角度に表示できる[バードビュー(鳥瞰図)]になります。メニューから表示/「バードビューの塗りつぶし」を実行してから、同じ表示メニュー/「バードビューのターン」を実行します。この状態でバードビュー上で自由にドラッグしてみてください。(FIG 2-2)

FIG 2-2



回転は表示のみで、モデル自体の回転処理は行われません。マウスを右クリックすると[パードビューのターン]が終了します。次に[立方体0]の頂点を移動して全体の厚みを変える編集を行いますが、その前に、現在の編集モードが[頂点選択]になっているかを確認しておきます。通常、DigitalLoca 起動直後は[頂点選択]モードになります。なお、[モデリングビュー]ウィンドウの大きさによって、メニューボタンのグループが2列表示になる場合もあります。(FIG 2-3)

次に、メニュー/コマンド/[範囲選択]を選びます。この操作は、メニューボタンやキーボードのショートカットキーからでも選択することが出来ます。ボタンの詳細はトピックを参照してください。(FIG 2-4)

カーソルが+の形になったら正面図上にて上の2つの頂点グループをドラッグで矩形選択します。(FIG 2-5)

選択された頂点は緑色になります。そのまま下方向にドラッグして、赤い中心線よりグリッド1つ上の位置まで移動させます。キーボードの下矢印[↓]キーでも操作できます。(FIG 2-6)

FIG 2-3

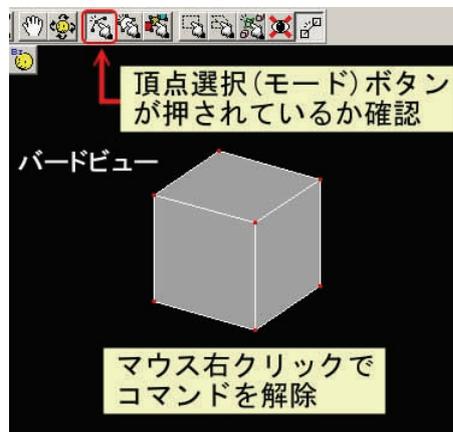


FIG 2-4



FIG 2-5

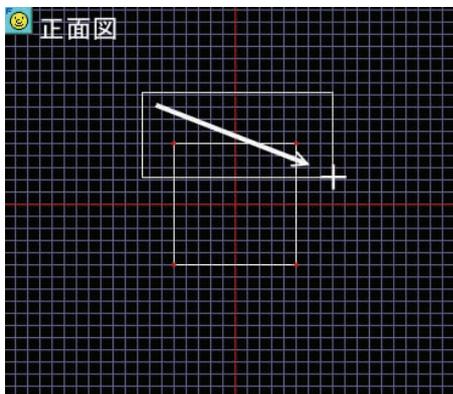
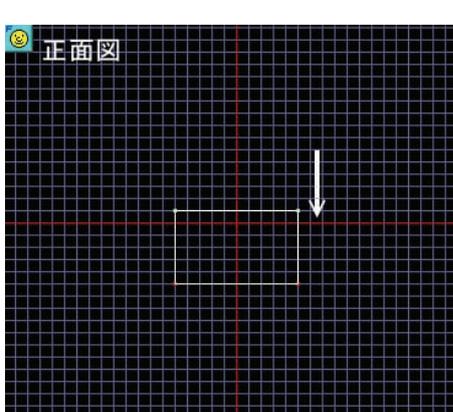


FIG 2-6



トピック

◆ モデリングビューの操作

モデリングビュー内の各操作には以下の3つの方法があります。

- メニューから選択
編集・表示・コマンド等
- メニューボタン
メニューの下にあるアイコンボタン
- ショートカットキー
キーボードのキー操作

◆ 頂点選択(モード)

ショートカットキー
'V'



◆ ポリゴン選択(モード)

ショートカットキー
'P'



◆ 範囲選択

ショートカットキー
'B'



◆ 正面図



トピック

◆ 範囲選択

ショートカットキー
‘B’



◆ 拡大表示

ショートカットキー
‘Shift’ + ‘I’



◆ ポリゴン選択(モード)

ショートカットキー
‘P’



再び [範囲選択] を実行し、同じく正面図下の2つの頂点グループを矩形選択します。前に選択した頂点グループは自動的に解除されます。

(FIG 2-7)

選択された頂点が緑色になったら、今度は上方向にドラッグして、赤い中心線よりグリッド1つ下の位置まで移動させます。キーボードの上矢印 [↑] キーでも操作できます。

(FIG 2-8)

頂点移動によるサイズの調整を終えたら、メニュー / 表示 / [拡大表示] (又は [拡大表示] ボタン) を実行して、画面を見

FIG 2-7

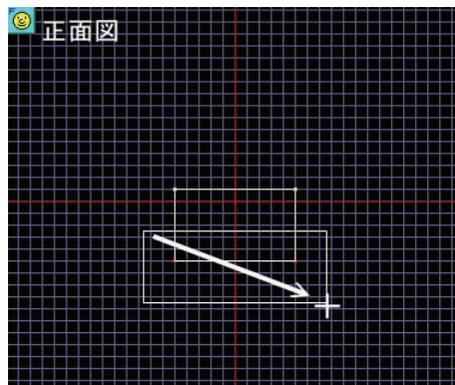


FIG 2-8

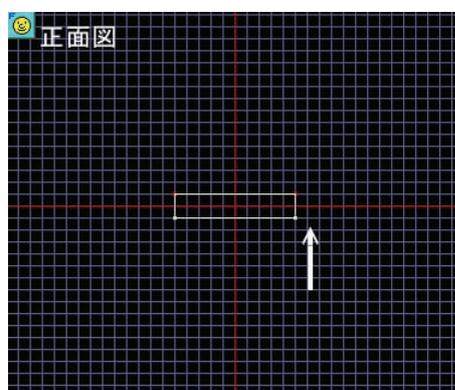
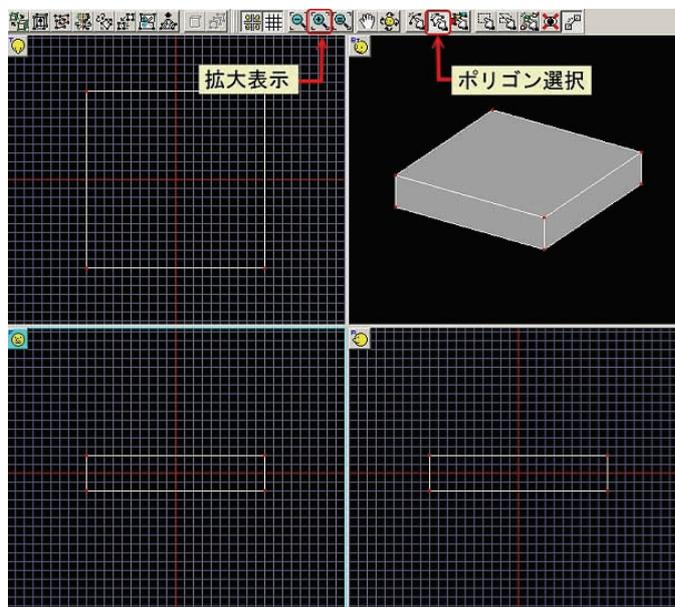
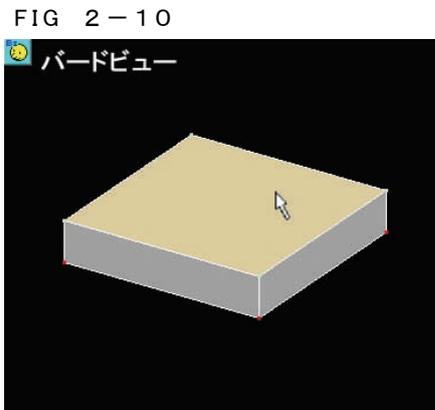


FIG 2-9

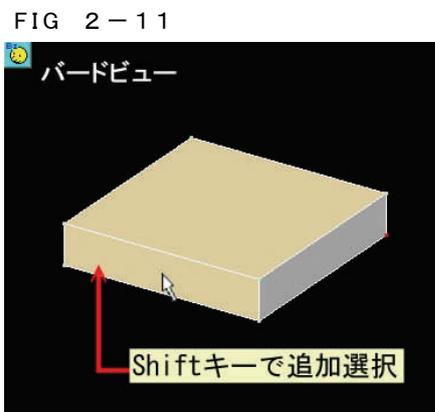


やすくします。次にモードを [ポリゴン選択] に変更します。[ポリゴン選択] モードでは、面単位での選択や加工編集が可能になります。これから [立方体0] をオフィスチェアの座面に加工していきますが、現在のシンプルな形状では編集要素が少ないので、ポリゴン(面)数を増やす処理を行います。(FIG 2-9)

パードビュー上で [立方体0] の上面が見えるよう表示 / [パードビューのターン] を実行して調節します。ターンを右クリックで解除してから、上面を 1 回クリックして選択します。(FIG 2-10)



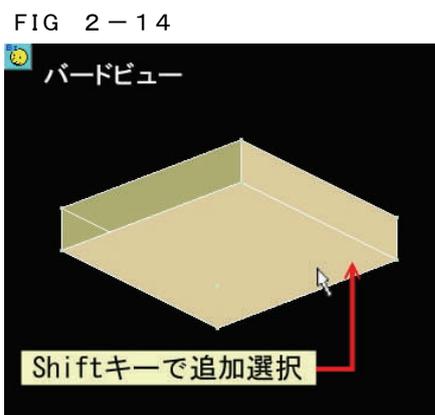
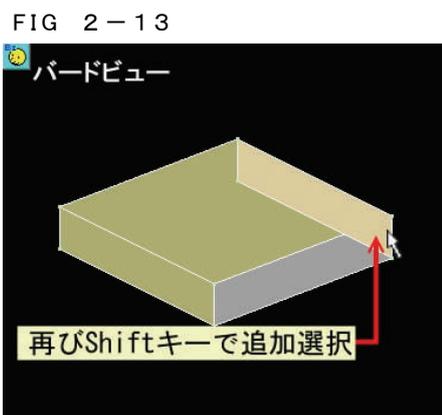
続いて [Shift] キーを押した状態で手前の面を追加選択します。(FIG 2-11)



反対の面を追加選択する為に、再び表示 / [パードビューのターン] を実行、カーソルを斜め左上にドラッグして裏側を表示した後、マウス右クリックでターンを終了させます。既に2つの面が選択されているので表示が見にくくなりますが、選択面が裏側(暗い色)になっていればOKです。(FIG 2-12)



再び [Shift] キーを押した状態で、残りのグレーの部分を1つずつ追加選択していきます。失敗した場合は [Space] キーで選択を解除してから、やり直してください。(FIG 2-13・14)



トピック

◆ パードビューのターン

ショートカットキー
'T'



(解除は右クリック)

トピック

◆ 選択ポリゴンの分割

ショートカットキー
‘X’



◆ 選択の解除

ショートカットキー
‘Space’

側面を除く面が筒状に選択されたら、コマンド / [選択ポリゴンの分割] を実行します。(FIG 2-15)

パードビュー上で任意の場所を1回クリックして、赤いポリゴン分割線を表示させます。(FIG 2-16)

キーボードの [Space] キーを押してポリゴン分割を決定します。もう一度 [Space] キーを押して選択状態も解除します。(FIG 2-17)

FIG 2-16



分割された[立方体0]の半分だけを更にポリゴン分割していきます。正面図上で[範囲選択]を実行し、左半分を矩形選択します。(FIG 2-18)

再び[範囲選択]を実行してから、[Shift]キーを押した状態で右の側面を矩形選択します。選択された状態はFIG 2-19のようになります。

FIG 2-15

コマンド(C)	
頂点選択(V)	V
●ポリゴン選択(P)	P
グループ選択(Q)	O
▼範囲選択(B)	B
領域選択(N)	N
共有頂点パーツの選択(G)	G
選択のみ表示(H)	H
●選択の移動(Q)	
ポリゴンの追加(A)	A
選択の回転(L)	L
選択の拡大・縮小(S)	S
選択の引き伸ばし(E)	E
掃引体の作成(F)	F
連続掃引体の作成(D)	D
回転体の作成(K)	K
選択ポリゴンの表裏反転(R)	R
選択ポリゴンの三角形変換(Y)	Y
選択ポリゴンの分割(X)	X
選択頂点の比例移動(U)	U
選択ポリゴンの四角形変換(O)	Shift+Y
選択ポリゴンの押しつぶし(L)	L

FIG 2-17

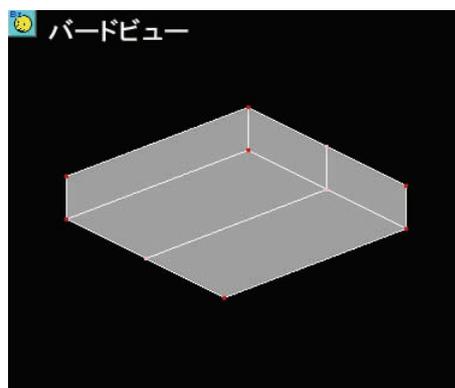


FIG 2-18

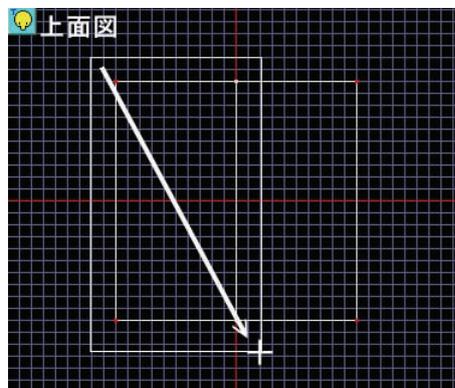
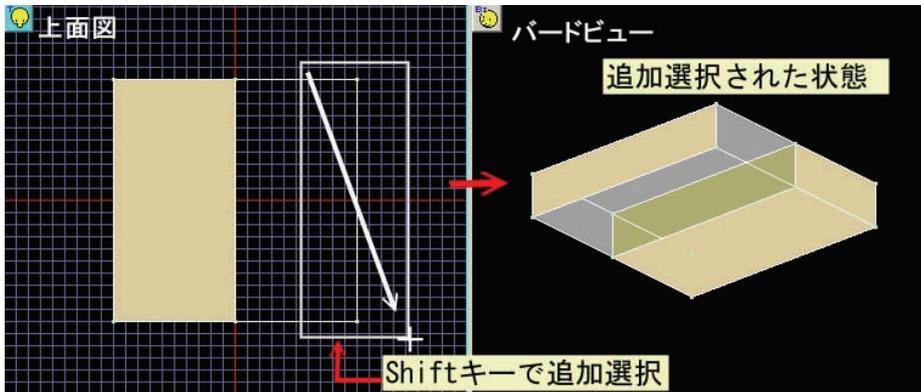


FIG 2-19



現在の選択部分を反転させる為に、メニューの中から編集/[選択状態の反転]を実行します。この処理で、側面を除く縦半分のポリゴンだけが選択されたこととなります。(FIG 2-20・21)

FIG 2-20

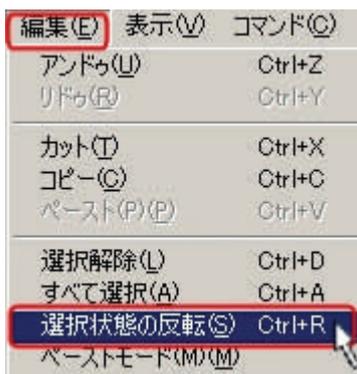
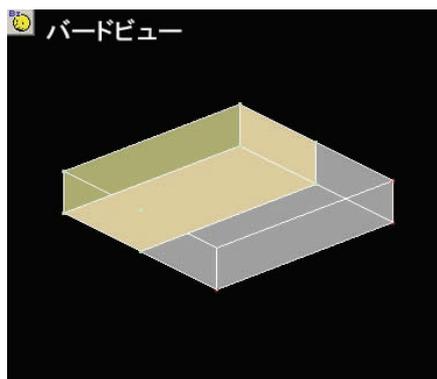


FIG 2-21



コマンド/[選択ポリゴンの分割]を実行して、バードビュー上の任意の場所を1回クリックして赤い分割線を表示させます。(FIG 2-22)

キーボードの[Space]キーを押して分割を決定、もう一度[Space]キーを押して選択状態を解除します。(FIG 2-23)

FIG 2-22

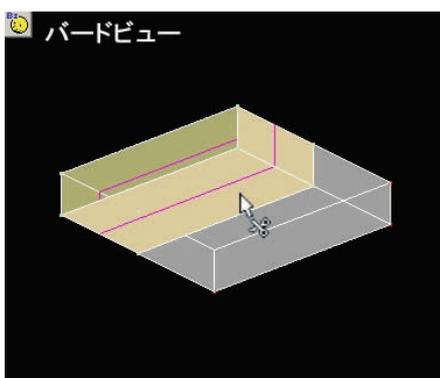
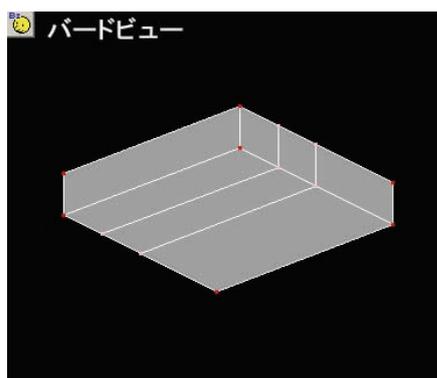


FIG 2-23



トピック

◆ 選択状態の反転

ショートカットキー
'Ctrl'+ 'R'

◆ 選択の方法(重要)

モデルの形状やポリゴンの密度に応じて、[Shift]キー+クリックによる追加選択や、[範囲選択]コマンドによる方法を適宜使い分けてください。

◆ 選択の部分解除(重要)

'Shift'キーを押しながら1度選択したポリゴンをクリックすると、その部分のみ解除されます。

◆ 選択ポリゴンの分割

ショートカットキー
'X'



トピック

◆ 頂点選択(モード)

ショートカットキー
'V'



◆ 範囲選択

ショートカットキー
'B'



◆ 縮小表示

ショートカットキー
'Shift' + 'O'



新たに作成された分割線を中央に移動する為に、[頂点選択](モード)に切り替えます。選択モードの切り替えは、モデリング編集で頻繁に使用するので、キーボードのショートカットキー(トピック参照)を覚えると便利です。

次にコマンド/[範囲選択]を実行して、上面図上から中央2つの分割線がある頂点グループを矩形選択します。(FIG 2-24)

選択された頂点グループを中央にドラッグして移動します。表示/[縮小表示]を実行して、次の編集作業に移ります。(FIG 2-25・26)

FIG 2-24

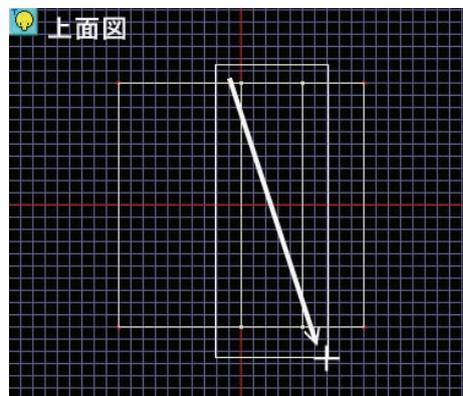


FIG 2-25

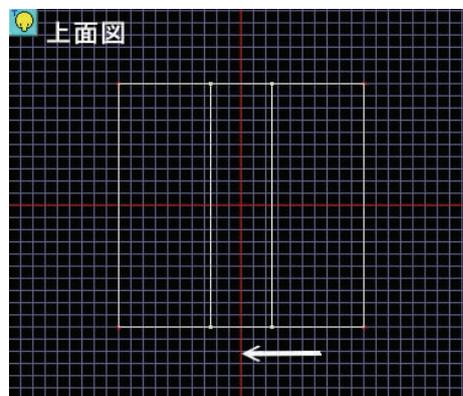


FIG 2-26

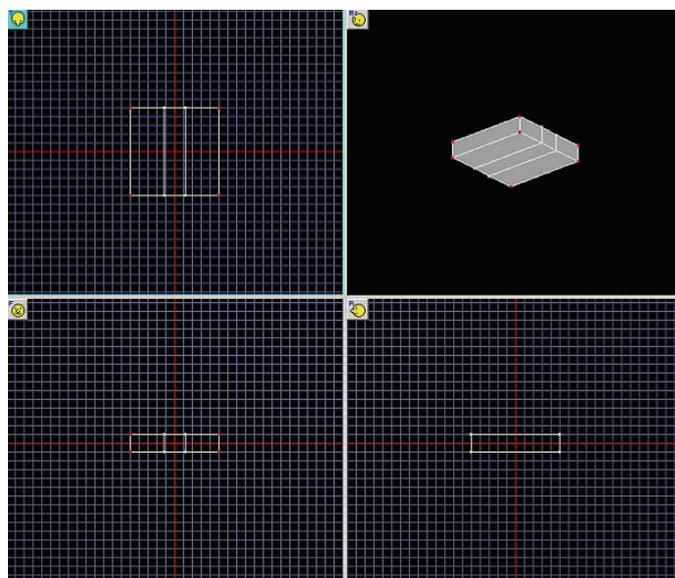
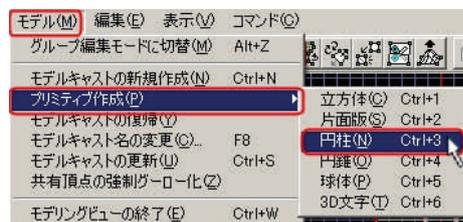


FIG 2-27

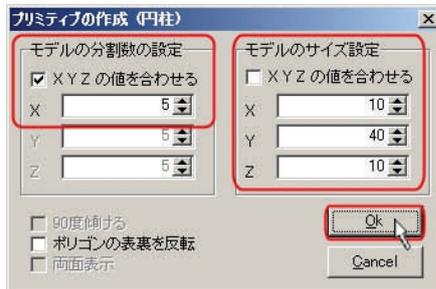


モデル/プリミティブ/[円柱]を選択して、[プリミティブの作成]パネルを表示させます。(FIG 2-27)

パネル内では円柱の分割数(3角形・8角形等)やサイズを任意に決定することができます。ここでは以下の設定にします。(FIG 2-28)

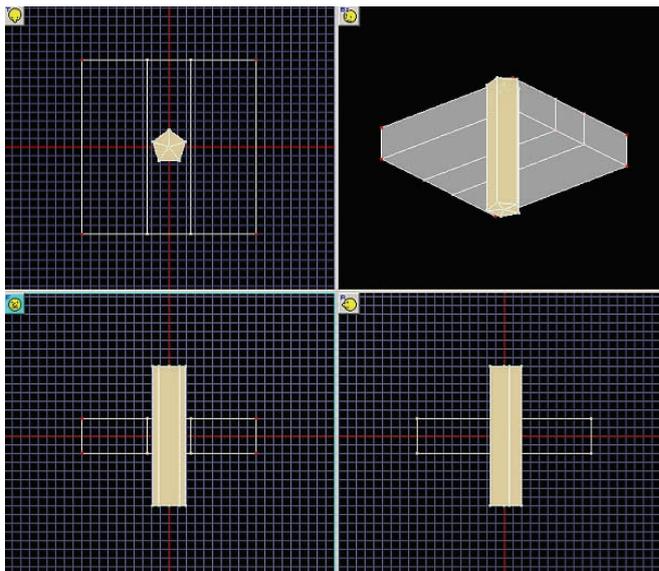
- モデルの分割数
X[5]
- モデルのサイズ設定
[XYZの値を合わせる]
にチェックをしない
X[10]・Y[40]・Z[10]

FIG 2-28



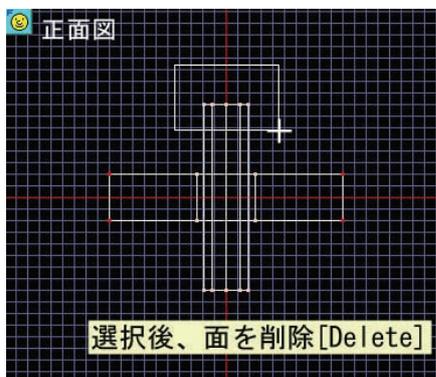
モデリングビュー内で作成されるプリミティブモデルは選択状態で追加されます。そのままモードを[ポリゴン選択]に切り替えます。(FIG 2-29)

FIG 2-29



この円柱はオフィスチェアの支柱になるので、不要な上下の面を削除しておきます。コマンド/[範囲選択]を実行、正面図上から上の面を矩形選択後、キーボードの[Delete]キーを押して削除します。選択や削除の状態はバードビュー内で確認しておきましょう。(FIG 2-30)

FIG 2-30



トピック

◆ ポリゴン選択(モード)

ショートカットキー
'P'



◆ 範囲選択

ショートカットキー
'B'



◆ 範囲選択の注意

[Shift]キーを併用しない限り、[選択範囲]コマンドを実行すると、既に選択された部分は解除されます。

◆ 削除

ショートカットキー
'Delete'

トピック

同様にして下の面も削除します。なお、[Shift] キーを併用した複数選択で上下の面を一度に削除してもかまいません。(FIG 2-31)

側面だけになった円柱を座面の下に移動する場合、円柱だけを選択する必要があります。現在のモデルはまだシンプルな形状なので、上面図上から[範囲選択]を使用することができます。しかし編集が進むにつれ頂点の重なり部分が多くなると、[範囲選択]だけでは一度に処理することが困難になります。今後の編集作業を考え、少し違ったアプローチで形状選択する方法を試してみましょう。

まず、円柱の任意のポリゴン(面)を1つだけクリックして選択します。(FIG 2-32)

次にコマンド/[共有頂点パーツの選択]を実行します。(FIG 2-33)

すると選択された面の頂点に共有する(連なる)全ての面が選択されます。この方法は、追加された形状や分断された一塊(かたまり)のパーツを選択するときに非常に便利です。(FIG 2-34)

FIG 2-31

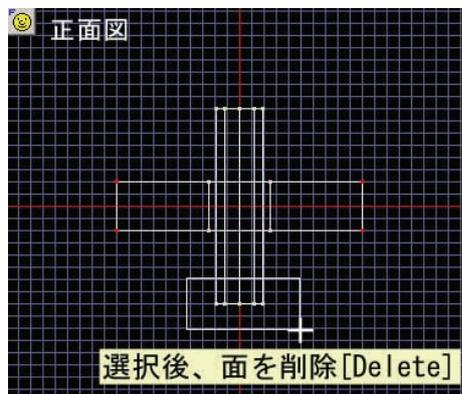


FIG 2-32

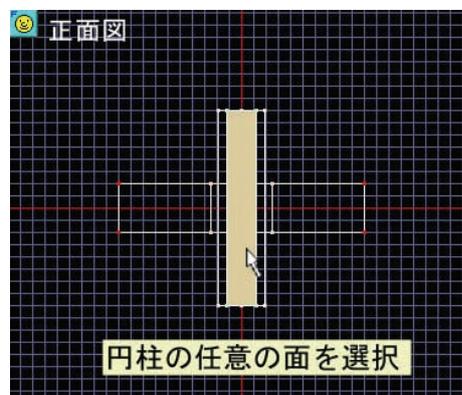
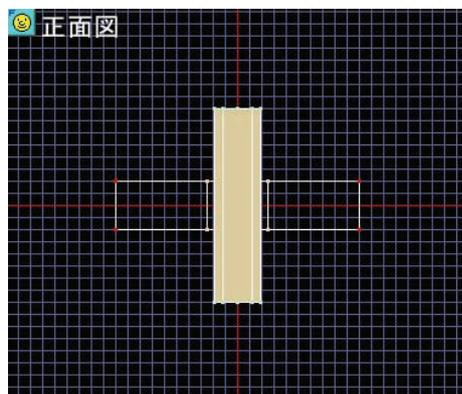


FIG 2-33



FIG 2-34



◆ 共有頂点パーツの選択

ショートカットキー
'G'



[共有頂点パーツの選択] コマンドは、[頂点選択] モードでも使用できます。その場合は任意の頂点を選択した後でこのコマンドを実行します。

選択状態になった円柱を座面の下にドラッグもしくはキーボードの下矢印[↓]キーで移動します。円柱の上部が立方体の底辺と重なるようにしてください。(FIG 2-35)

選択状態のまま、コマンド/[選択のみ表示]を実行します。(FIG 2-36)

すると選択状態の円柱のみ表示され、他は非表示になります。このコマンドを使うと表示部分のみが編集の対象となります。(FIG 2-37)

FIG 2-36



FIG 2-35

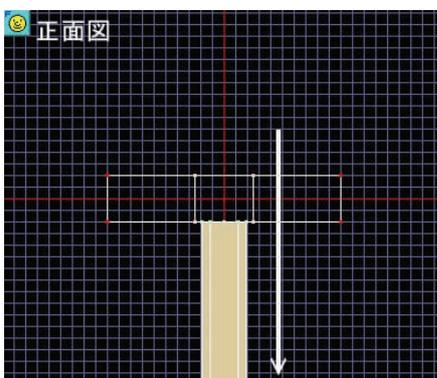
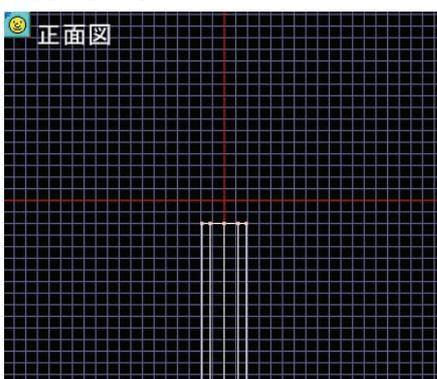


FIG 2-37

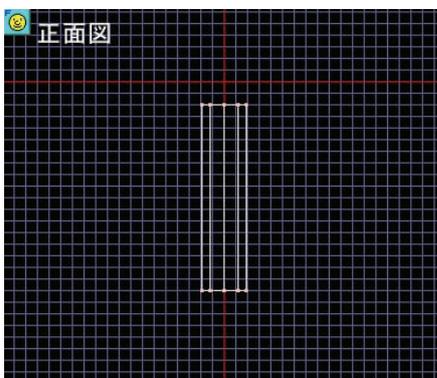


正面図では円柱の下側が隠れているので編集作業に不都合です。コマンド/[表示サイズの自動調整]を実行して中央に表示させます。(FIG 2-38・39)

FIG 2-38



FIG 2-39



トピック

◆ 選択のみ表示

ショートカットキー
‘H’



◆ 表示サイズの自動調整

ショートカットキー
‘Ctrl’ + ‘F’

*このコマンドには、ボタンはありません。

トピック

◆ すべて選択

ショートカットキー
‘Ctrl’ + ‘A’

◆ 選択ポリゴンの分割

ショートカットキー
‘X’



◆ 選択の解除

ショートカットキー
‘Space’

◆ 頂点選択(モード)

ショートカットキー
‘V’



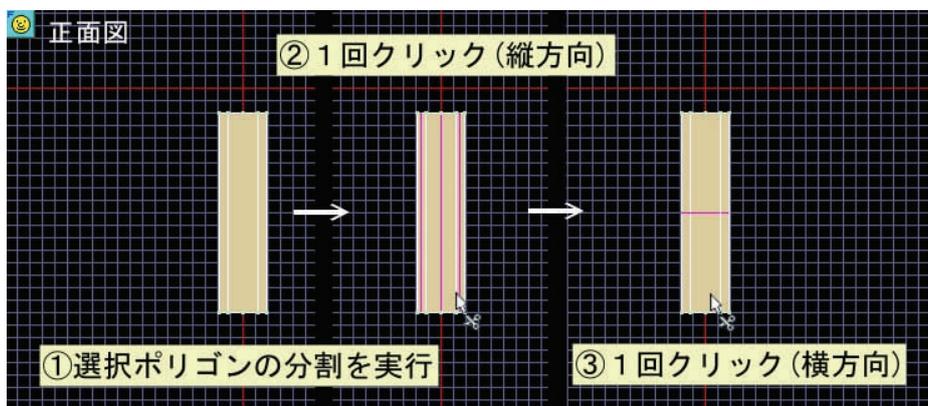
◆ 範囲選択

ショートカットキー
‘B’



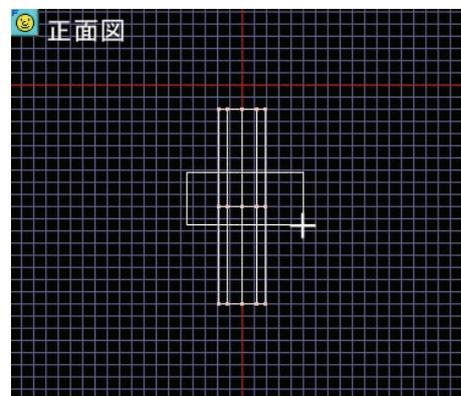
円柱を分割してオフィスチェアの足を追加します。まず、編集 / [すべて選択] を実行して選択状態にします。次にコマンド / [選択ポリゴンの分割] を実行して任意の面を1回クリックします。分割線はクリックごとに [縦方向] - [横方向] - [なし] - [縦方向] を繰り返し、[Space] キーで決定しない限り何回でもやり直すことができます。分割線が横方向になったら、キーボードの [Space] キーを1回押して分割を決定、もう一度 [Space] キーを押して選択を解除します。(FIG 2-40)

FIG 2-40



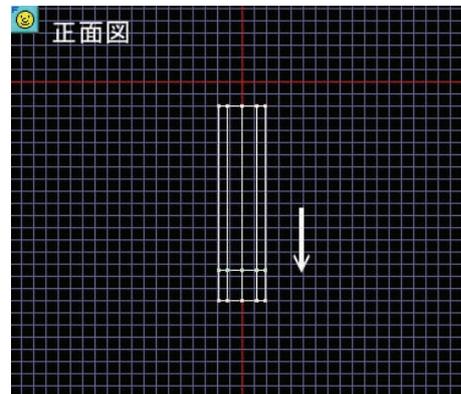
[頂点選択]モードに切り替え、コマンド[範囲選択]を実行します。正面図上から円柱中央の分割頂点グループを矩形選択します。(FIG 2-41)

FIG 2-41



そのまま円柱の下側へ、ちょうどグリッド2つ半ほど離れた位置にドラッグで移動します。キーボードの下矢印[↓]キーで操作すると正確に移動できます。(FIG 2-42)

FIG 2-42



[ポリゴン選択] モードに切替え、正面図上から円柱の下側正面のポリゴンをクリックして選択します。(FIG 2-43)

コマンド / [選択の引き伸ばし] を実行します。(FIG 2-4

FIG 2-43

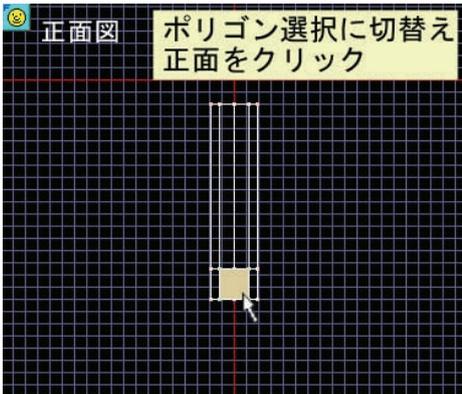


FIG 2-44



4)

選択状態のポリゴンを上面図上で下方向にビューの境界線近くまでドラッグします。 ボタンを離すと自動的に新しいポリゴン形状が作成されます。 [引き伸ばし] 直後は自動的に [頂点選択] モードになり、先端部分が選択状態になることに注意してください。

(FIG 2-45・46)

FIG 2-45

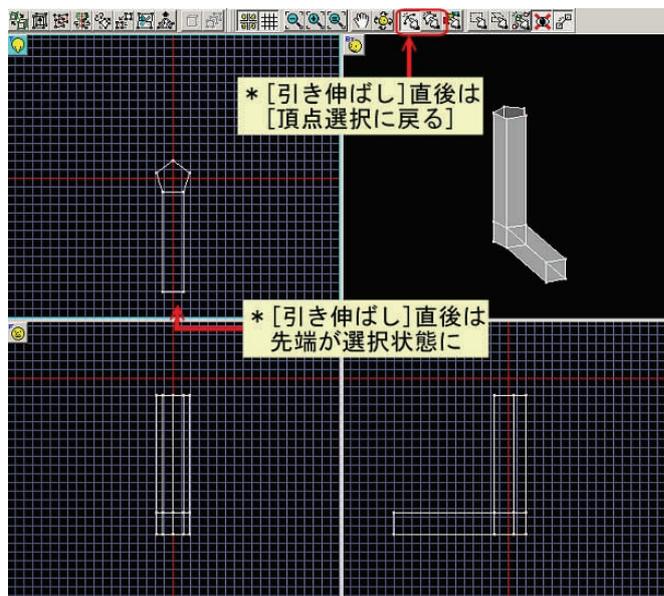
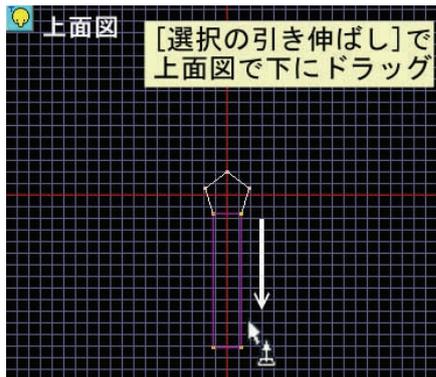


FIG 2-46

トピック

◆ ポリゴン選択(モード)

ショートカットキー
'P'



◆ 選択の引き伸ばし

ショートカットキー
'E'



トピック

◆ バードビューのターン

ショートカットキー
‘T’



(解除は右クリック)

◆ ポリゴン選択(モード)

ショートカットキー
‘P’



◆ 選択の引き伸ばし

ショートカットキー
‘E’



選択状態をキーボードの [Space] キーで解除した後、バードビュー上に移り、引き伸ばされたポリゴンの左側が見えるように [バードビューのターン] で表示調整をします。再び [ポリゴン選択] モードに切替え、円柱の左下のポリゴン面をクリックして選択します。(FIG 2-47)

コマンド/[選択の引き伸ばし] を実行、上面図上にて斜め左下方向にドラッグで引き伸ばします。位置は先に引き伸ばしたポリゴン形状を参考にしてください。(FIG 2-48)

新たなポリゴンが作成されたら、選択状態をキーボードの [Space] キーで解除します。バードビュー上に移り、今度は円柱の右側が見えるように [バードビューのターン] で表示調整をします。再び [ポリゴン選択] モードに切替え、円柱右下のポリゴン面をクリックして選択します。(FIG 2-49)

コマンド/[選択の引き伸ばし] を実行、上面図上にて斜め右下方向にドラッグで引き伸ばします。(FIG 2-50)

FIG 2-47

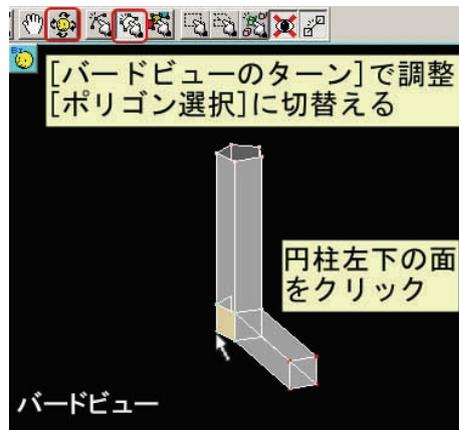


FIG 2-48

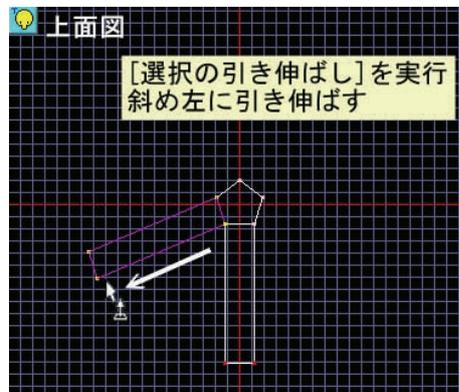


FIG 2-49

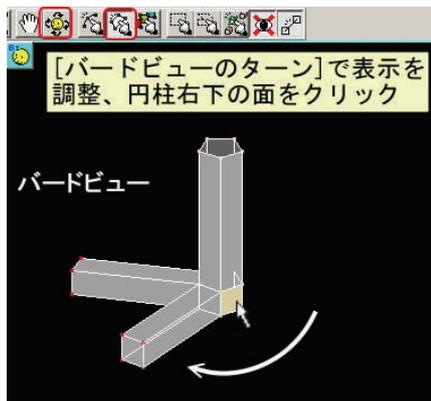
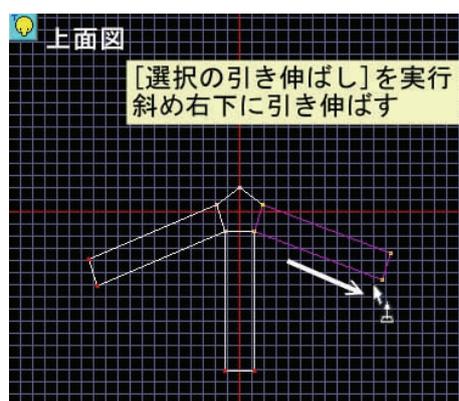


FIG 2-50



最近のオフィスチェアは5脚が主流なので、円柱を五角形にしたわけですが、[選択の引き伸ばし]コマンドで引き伸ばす際、角度や長さは、座面を再表示させてから行います。

引き続き、残りの脚の引き伸ばし作業を行きましょう。選択状態を[Space]キーで解除してから、バードビューの表示を円柱の残りの面が見えるように[バードビューのターン]で調整します。必ず[ポリゴン選択]モードに切替えてから、残りのポリゴン面を選択します。コマンド/[選択の引き伸ばし]で作成する手順は今までと同じです。(FIG 2-51・52)

5脚全てが完成したら、コマンド/[選択のみ表示]を実行します。このコマンドはトグル(On/Off)スイッチになっていて、再度実行すると解除され、隠されていたポリゴンが表示されます。(FIG 2-53)

現在のモードは[頂点選択]になっているはずですが、一応確認しておきます。コマンド/[範囲選択]を実行して上面図上から左上の脚先端部を矩形選択します。選択後、長さや角度の修正が必要であれば移動調整します。(FIG 2-55)

脚部は支柱に対して傾斜角

FIG 2-51

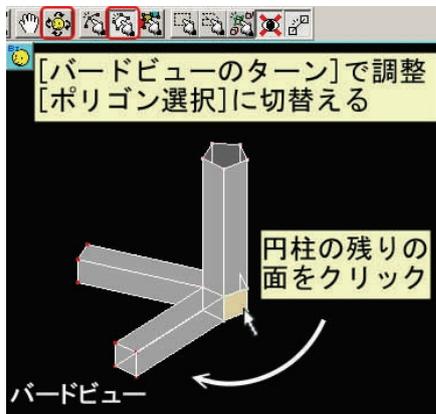


FIG 2-52

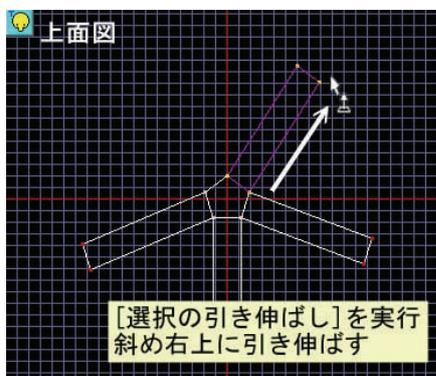
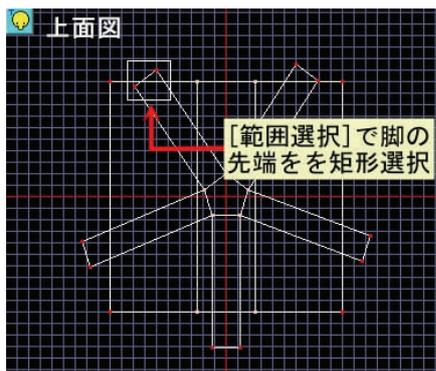


FIG 2-53



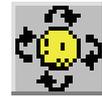
FIG 2-54



トピック

◆ バードビューのターン

ショートカットキー
'T'



(解除は右クリック)

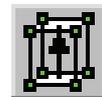
◆ ポリゴン選択(モード)

ショートカットキー
'P'



◆ 選択の引き伸ばし

ショートカットキー
'E'



◆ 頂点選択(モード)

ショートカットキー
'V'



◆ 選択のみ表示

ショートカットキー
'H'



◆ 範囲選択

ショートカットキー
'B'



トピック

◆ 範囲選択

ショートカットキー
‘B’



度を持って取り付けられています。そこで選択状態のまま正面図に移り、下方方向に少しだけドラッグ移動させて傾斜をつけます。

(FIG 2-55)

上面図に戻り、コマンド/[範囲選択]を実行して右上の脚先端部を矩形選択します。座面端の頂点と接近しているので注意してください。この部分も長さや角度の修正が必要であれば移動調整します。(FIG 2-56)

選択状態のまま正面図に移り、左脚の形状角度と対称になるように下方方向にドラッグ移動します。(FIG 2-57)

5脚の角度調整を終えるとFIG 2-58のようになります。

チュートリアルツアーでは、モデリング操作のナビゲーション習得をテーマに解説している関係上、モデル形状がシンプルなものになっています。オリジナルで独創的な形状—例えば脚部分の湾曲等にチャレンジしたいのであれば、これまでに解説した[選択の引き伸ばし]を脚の先端部に繰り返し実行することで表現できます。色々な手法を組み合わせ工夫してみてください。

FIG 2-55

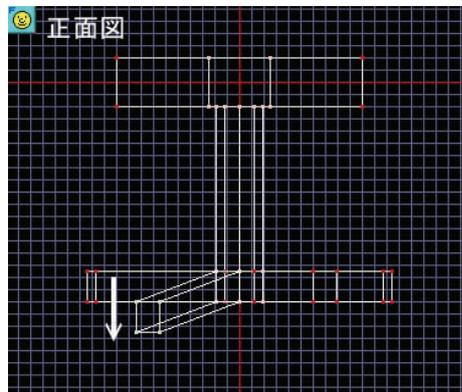


FIG 2-56

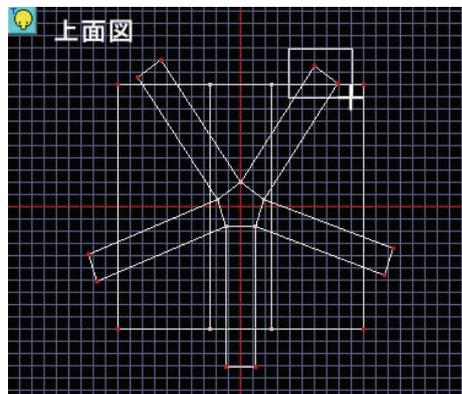


FIG 2-57

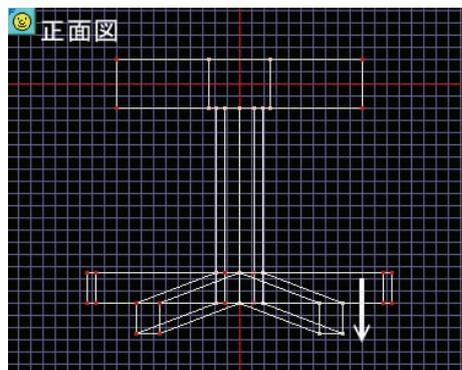
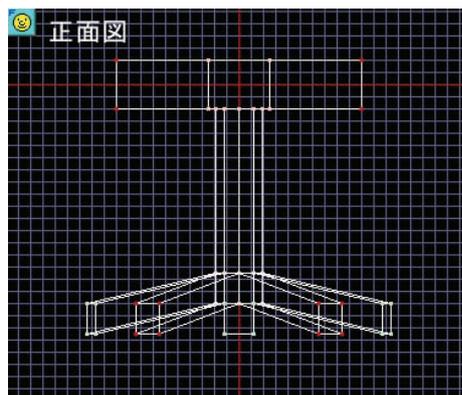
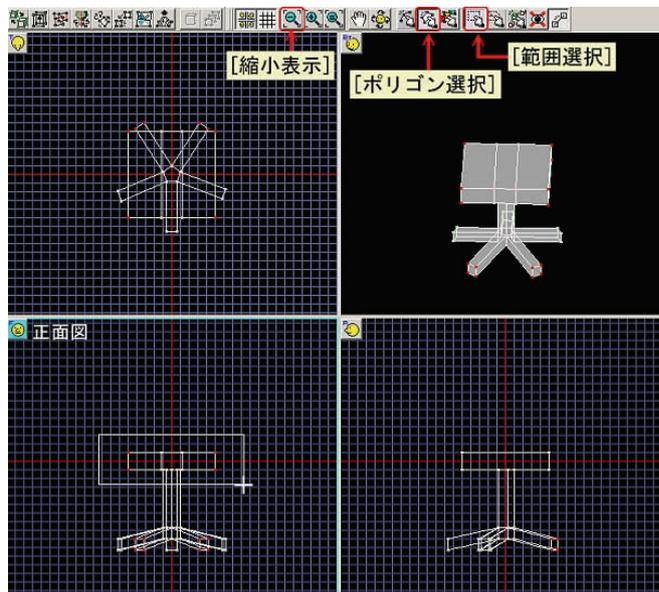


FIG 2-58



表示 / [縮小表示] を実行してから、モードを [ポリゴン選択] に切替えます。 コマンド / [範囲選択] を実行して、正面図上で座面部分のポリゴンを矩形選択します。(FIG 2-59)

FIG 2-59



座面全てが選択されたら、編集 / [コピー] を実行します。(FIG 2-60・61)

FIG 2-60

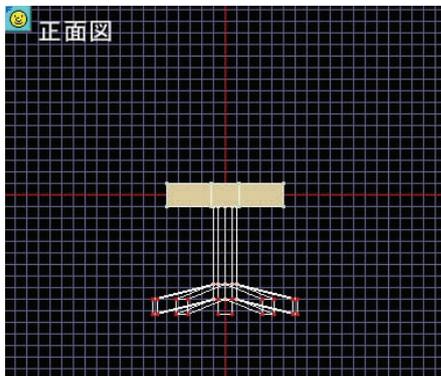


FIG 2-61

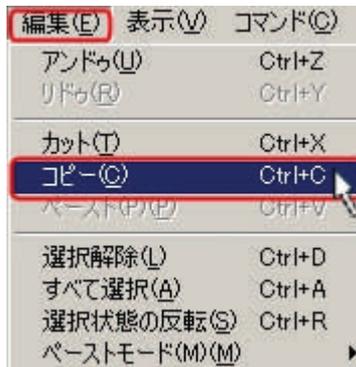


FIG 2-62



複製した座面をチェアの背もたれにしようと思いましたが、ペースト(貼り付け)する前に、編集 / [ペーストモード] を [コピー時のグループ] に設定しておきます。この [ペーストモード] に関しては後で解説します。(FIG 2-62)

トピック

◆ 縮小表示

ショートカットキー
'Shift' + 'O'



◆ ポリゴン選択(モード)

ショートカットキー
'P'



◆ 範囲選択

ショートカットキー
'B'



◆ コピー

ショートカットキー
'Ctrl' + 'C'

トピック

◆ ペースト

ショートカットキー
‘Ctrl’ + ‘V’

◆ 選択の回転

ショートカットキー
‘L’



◆ X座標

モデリングビューの正面図や上面図から見て横方向に伸びるライン。ちなみに[Y座標]は正面図や側面図から見て縦方向のライン、[Z座標]は上面図から見て縦方向、側面図から見て横方向のラインになります。

◆ モデルの座標

モデリングビュー内の座標は、モデル自身が持つ性格のような座標で、[ローカル座標]といいます。ローカル座標は中心軸がどこにあるのかを定義します。3D空間内のどこに配置されても、このローカル座標の中心軸を基準に回転や拡大をすることになります。

編集/[ペースト](貼り付け)を実行します。複製された座面は当然同じ位置にペーストされるので、背もたれに相応しい角度調整を行います。選択状態のまま、コマンド/[選択の回転]を実行してください。(FIG 2-64)

FIG 2-63

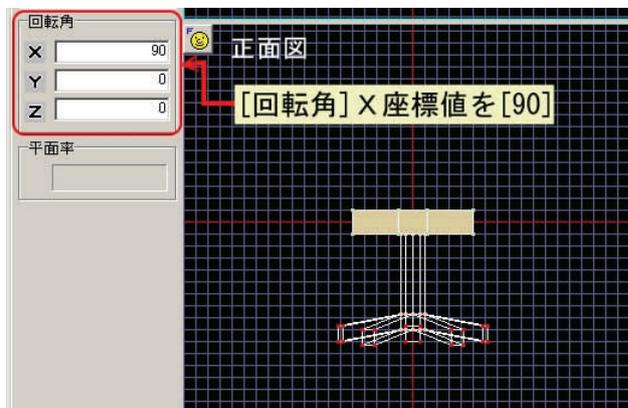
編集(E)	表示(V)	コマンド(C)
アンドゥ(U)	Ctrl+Z	
リドゥ(R)	Ctrl+Y	
カット(T)	Ctrl+X	
コピー(C)	Ctrl+C	
ペースト(P)	Ctrl+V	
選択解除(L)	Ctrl+D	
すべて選択(A)	Ctrl+A	
選択状態の反転(S)	Ctrl+R	
ペーストモード(M)	(M)	

FIG 2-64

コマンド(C)	
頂点選択(V)	V
ポリゴン選択(P)	P
グループ選択(O)	O
範囲選択(B)	B
領域選択(N)	N
共有頂点パーツの選択(G)	G
選択のみ表示(H)	H
選択の移動(M)	
ポリゴンの追加(A)	A
選択の回転(L)	L
選択の拡大・縮小(S)	S
選択の引き伸ばし(E)	E
掃引体の作成(F)	F

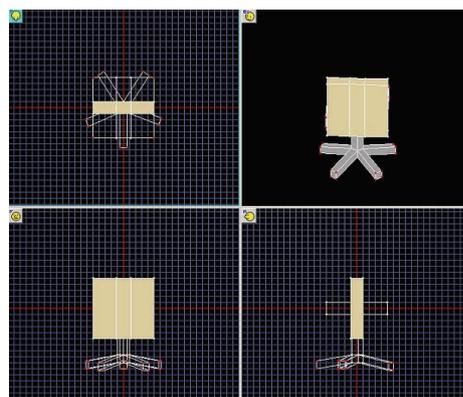
[選択の回転]が実行されると、モデリングビュー左下にある数値パネルが[回転角]の入力欄に変化します。X座標の回転角値にキーボードから[90]を入力して[Enter]キーを押します。(FIG 2-65)

FIG 2-65



コピーされた座面がX座標軸に対して90度回転しました。回転は座標の中心=赤いラインを基準に行われるのでオリジナルの座面と交差しています。背もたれに相応しい位置まで移動してみましょう。(FIG 2-66)

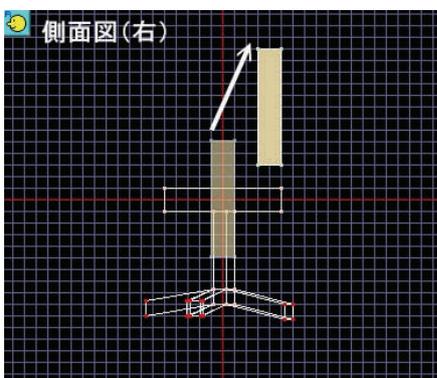
FIG 2-66



選択状態のまま、横を向いた顔のアイコンがある[側面図(右)]上で、座面の右上までドラッグで移動します。座面からの高さはグリッド2つ分くらいの隙間が目安です。移動が終了したら[Space]キーで選択状態を解除します。

(FIG 2-67)

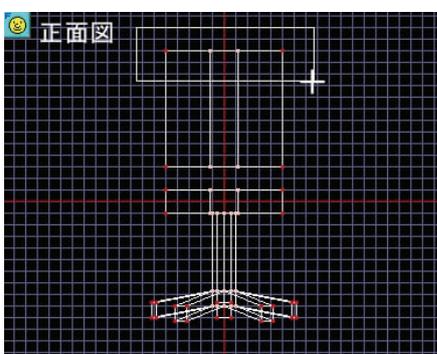
FIG 2-67



次に背もたれの頂点を編集して形状を整えます。[頂点選択]モードに切替えてから、コマンド/[範囲選択]を実行し、正面図上にて背もたれ上部の頂点を矩形選択します。

(FIG 2-68)

FIG 2-68



オフィスチェアとしてのバランスを考えながら、選択された頂点を下方向にドラッグ移動します。以下各コーナーや幅を[範囲選択]とドラッグ移動を使用して整えます。(FIG 2-69)

側面図(右)上からも厚みや角度を調整します。(FIG 2-70)

FIG 2-69

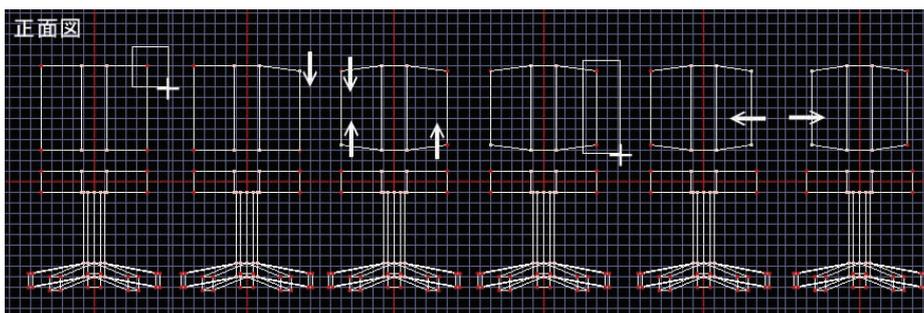
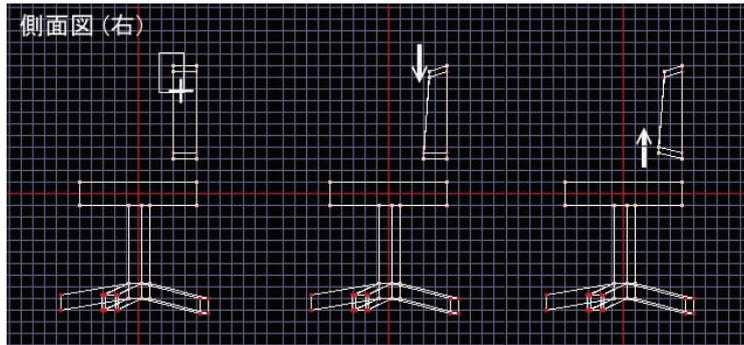


FIG 2-70



トピック

◆ 側面図(右)



◆ 頂点選択(モード)

ショートカットキー
'V'



◆ 範囲選択

ショートカットキー
'B'



トピック

◆ 範囲選択

ショートカットキー
‘B’



◆ 中心点の変更

ショートカットキー
‘Alt’ + 右クリック
この操作は形状編集時のみ有効で、本来のモデル中心位置はブルーラインで表示されます。モデル自身の中心を変更して保存したい場合は、モデル全体を選択して移動させます。

◆ 選択の回転

ショートカットキー
‘L’



◆ [Shift]+マウスドラッグ操作による回転方法

この操作は特定の座標軸方向にだけ回転させたい時に使用します。回転は各ビューの中心ラインを基準に行われます。縦方向にドラッグすると横の中心ラインを基準に回転し、横方向にドラッグすると縦の中心ラインを基準に回転します。縦と横のラインの座標は操作するビューによってX・Y・Zが変わるので注意してください。FIG 2-73の例では、横がX座標軸、縦がZ座標軸となります。

背もたれの形状編集を終えましたが、オフィスチェアらしい若干の傾斜が欲しいところです。しかし現在の中心点はモデル全体の中央にあります。そこで現在の中心点を一時的に変更して、背もたれの角度のみ調整することになります。まず、コマンド/[範囲選択]で背もたれ部分だけ矩形選択します。

(FIG 2-71)

次に[Alt]キーを押しながら、側面図上にて背もたれの右下の角を右クリックします。この操作で編集時のみ有効な中心点の変更が可能になります。(FIG 2-72)

コマンド/[選択の回転]を実行しますが、今度は数値入力ではなくマウス操作による任意の角度調整を行います。カーソルが回転を示す形状に変化したら、[Shift]キーを押しながら上面図で上方向にドラッグします。回転の様子は側面図で確認できるのでドラッグを維持したまま角度調整してください。(FIG 2-73)

FIG 2-71

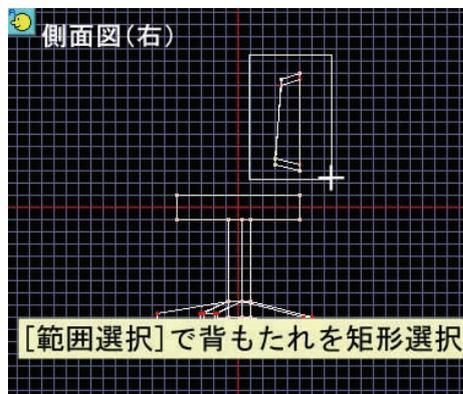


FIG 2-72

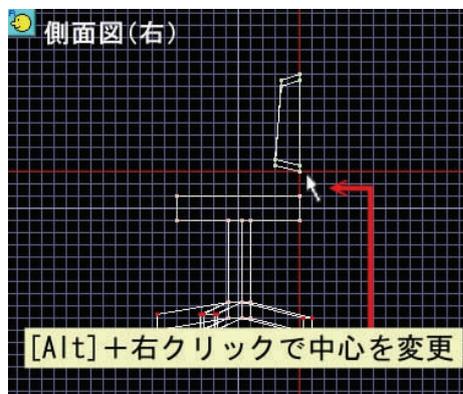
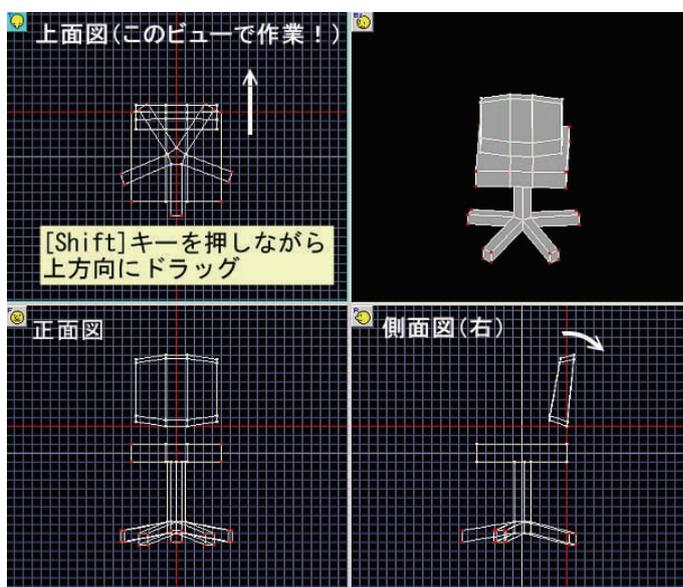


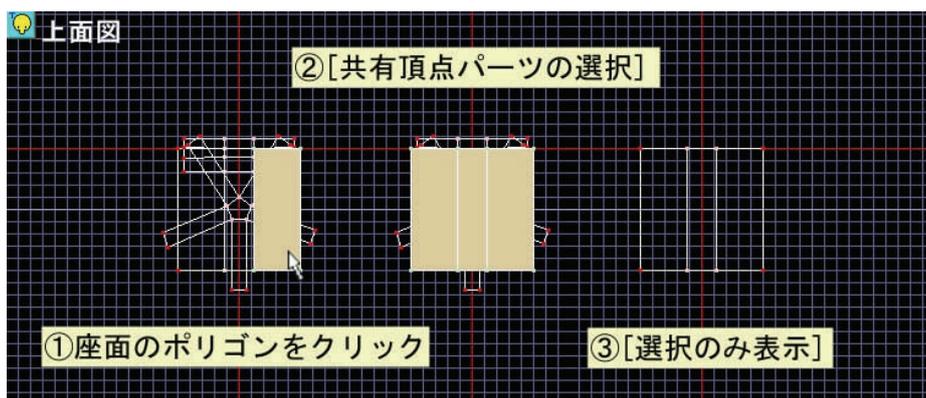
FIG 2-73



(重要！)
回転コマンドは数値入力、ドラッグによる任意操作とも1回のみ有効です。失敗した場合は、編集/[アンドゥ]でやり直し、再びコマンド/[選択の回転]を実行してください。

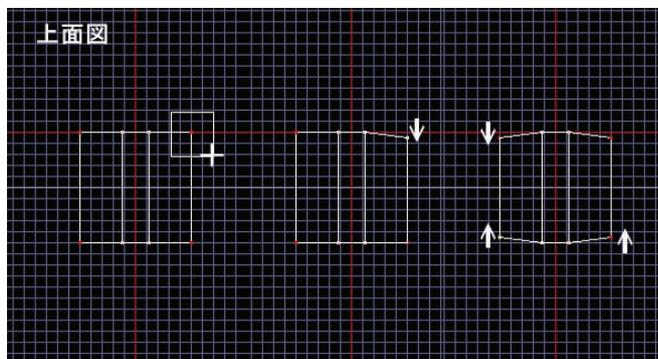
かなりオフィスチェアらしくなってきました。座面の方も少し修正を加えます。座面のみを編集対象にする為、いったん[ポリゴン選択]モードに切替え、上面図上で座面のポリゴン(任意)を1つクリックします。コマンド/[共有頂点パーツの選択]を実行して座面全てを選択状態にします。コマンド/[選択のみ表示]を実行し座面のみ表示させます。(FIG 2-74)

FIG 2-74



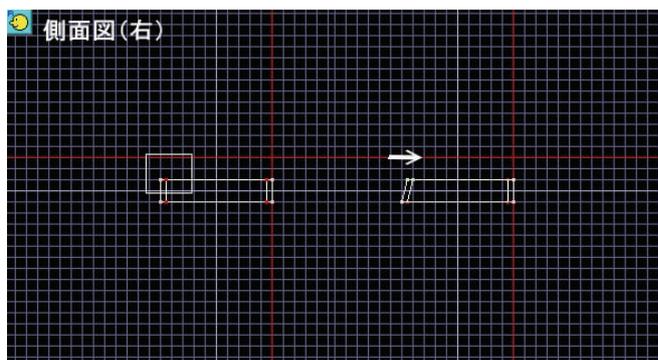
再び[頂点選択]モードに切替え、座面の各コーナーを個別に[範囲選択]して角を修正します。(FIG 2-75)

FIG 2-75



側面図(右)上からも角の修正を行います。(FIG 2-76)

FIG 2-76



トピック

◆ ポリゴン選択(モード)

ショートカットキー
'P'



◆ 共有頂点パーツの選択

ショートカットキー
'G'



◆ 選択のみ表示

ショートカットキー
'H'



◆ 頂点選択(モード)

ショートカットキー
'V'



トピック

◆ 選択のみ表示

ショートカットキー
‘H’



◆ 拡大表示

ショートカットキー
‘Shift’ + ‘I’



◆ 視点のスライド移動

ショートカットキー
‘M’



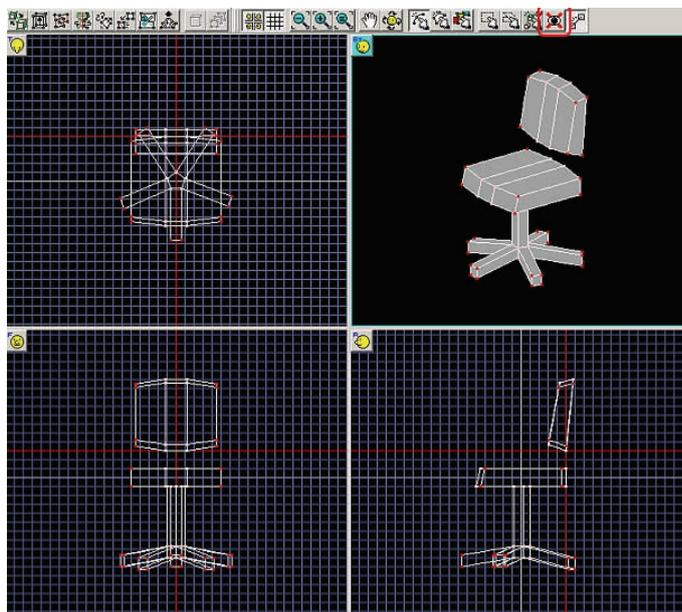
◆ ポリゴンの追加

ショートカットキー
‘A’



形状修正を終えたら、コマンド / [選択のみ表示] を再度実行して全てを表示させます。ボードビューで全体のバランスを確認してください。(FIG 2-77)

FIG 2-77



座面と背もたれをつなぐパーツは、0から新規ポリゴンを追加する方法で作成します。まず、表示 / [拡大表示] を実行します。次に表示 / [視点のスライド移動] で上面図の右側にスペースが空くように、表示を斜め左下に移動します。この操作は表示のみでモデル自体が移動するわけではありません。スライド移動の終了はマウス右クリックで行います。

(FIG 2-78)

コマンド / [ポリゴンの追加] を実行します。

(FIG 2-79)

FIG 2-78

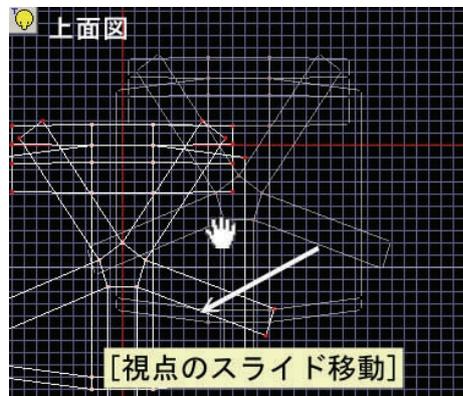
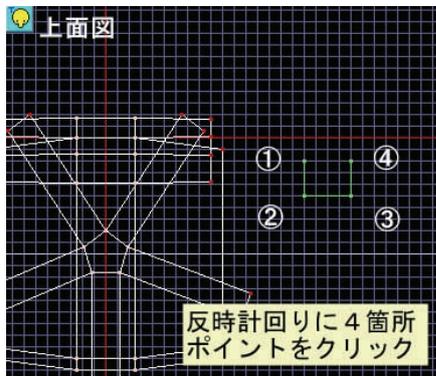


FIG 2-79

コマンド(C)	
● 頂点選択(V)	V
● ポリゴン選択(P)	P
● グループ選択(O)	O
✓ 範囲選択(B)	B
● 領域選択(N)	N
● 共有頂点パーツの選択(G)	G
● 選択のみ表示(H)	H
● 選択の移動(M)	M
● ポリゴンの追加(A)	A
● 選択の回転(L)	L
● 選択の拡大・縮小(S)	S
● 選択の引き伸ばし(E)	E

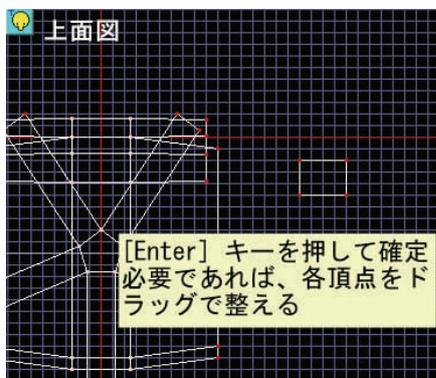
カーソルが鉛筆のような形状に変わったら、右側のスペース上でパーツの断面図を反時計回りに4箇所のポイントをクリックして作成します。断面図のラインは緑色で表示されます。5箇所以上クリックしてしまったら、間違えた場合は [Escape] キーを押してやり直してください。(FIG 2-80)

FIG 2-80



[Enter] キーを押すと、4つのポイントが繋がり、新規ポリゴンが作成されます。形状が歪んでいる時は、一旦 [頂点選択(モード)] にしてから、各頂点を個別にドラッグして調整してください。(FIG 2-81)

FIG 2-81



(頂点が揃わない時は！)
[頂点選択(モード)]でドラッグによる頂点移動をする時、初期設定では、[グリッドに合わせる]機能が働いてグリッド半分単位での移動になります。自由に少しずつ移動させたい時は、メニュー/表示/[グリッドに合わせる]('Ctrl'+ 'J')を選択して機能を解除してください。

再び [ポリゴン選択(モード)] にして、作成された面をクリック選択します。

(FIG 2-82)

続いて、コマンド/[ポリゴンの表裏反転]を実行してから選択された面をクリックします。

(FIG 2-83)

FIG 2-82

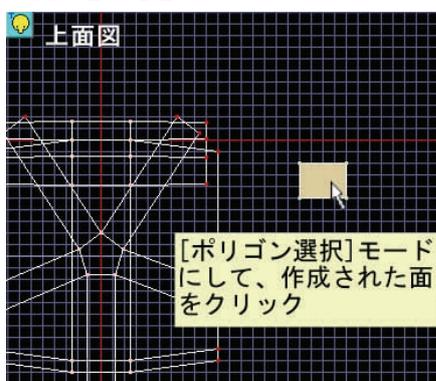
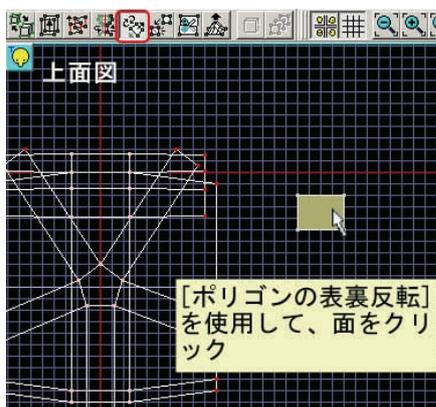


FIG 2-83



トピック

◆ (ポリゴン追加時の注意)

ポリゴンを追加する時のポイントは4箇所、始めのポイントに繋げる必要はない。四角形のポリゴンは4つの頂点ポイントで構成される。

(3角形ポリゴンを作成する場合は、3つのポイントで [Enter] キーを押す。)

◆ 頂点選択(モード)

ショートカットキー
'V'



◆ ポリゴン選択(モード)

ショートカットキー
'P'



◆ ポリゴンの表裏反転

ショートカットキー
'R'



トピック

◆ 側面図(右)



◆ 選択の引き伸ばし

ショートカットキー
‘E’



◆ 範囲選択

ショートカットキー
‘B’



選択されたポリゴンが裏側(暗い黄色)になったら、上面図から見て、背もたれの支柱が固定されている位置(座面の中央より上側)を想定してドラッグで移動します。(FIG 2-84)

ポリゴンが選択状態のまま側面図上に移り、コマンド/[選択の引き伸ばし]を実行し、下方方向に厚みがでるよう少しだけドラッグします。(FIG 2-85)

[選択の引き伸ばし]終了後は自動的に[頂点選択(モード)]になります。背もたれの支柱を続けて引き伸ばす為、切り口角度を編集してみましょう。引き伸ばされた右下の頂点グループを、コマンド/[範囲選択]を実行して囲みます。(FIG 2-86)

切り口が45度になるように選択された頂点グループを上へドラッグ移動します。この時、座面に重ならないように注意します。近づけすぎると後の選択作業ができなくなることがあります。(FIG 2-87)

ここで[選択の引き伸ばし]編集作業における[ポリゴン選択(モード)]と[頂点選択(モード)]の使い分けについて説明します。これまでの作業のように、新規に面を立体

FIG 2-84

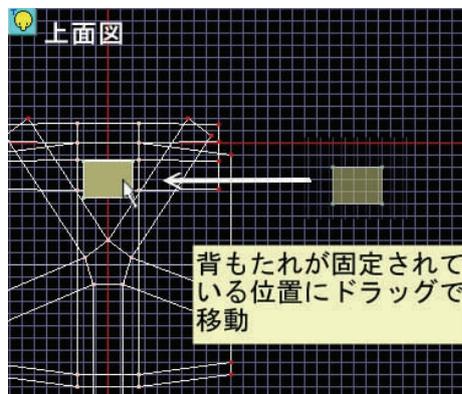


FIG 2-85

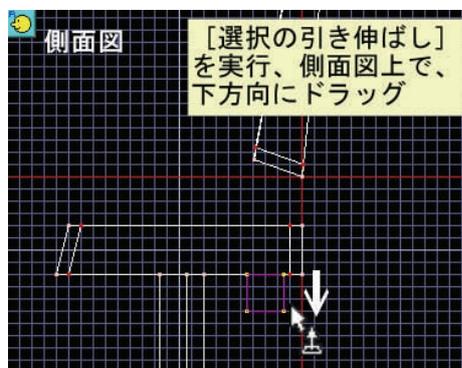


FIG 2-86

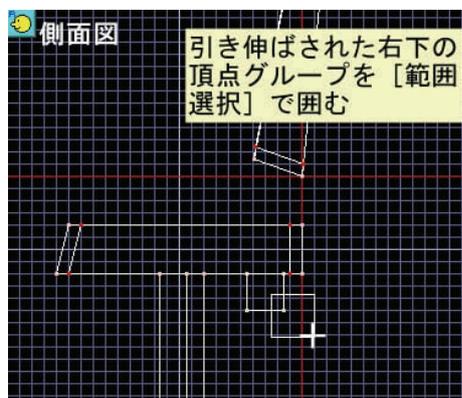
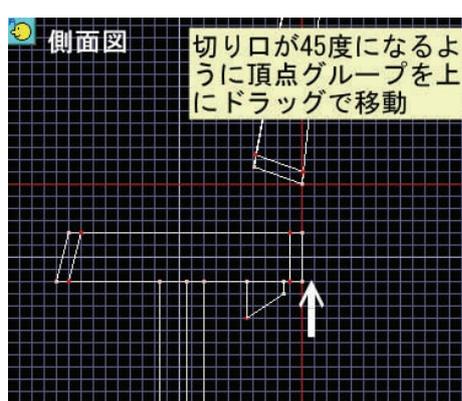


FIG 2-87



化する場合は、[ポリゴン選択(モード)]で対象のポリゴンを選択してからコマンド/[選択の引き伸ばし]を実行します。一度立体化された「部品」をさらに引き伸ばす場合は[頂点選択(モード)]で対象の頂点グループを選択してからコマンド/[選択の引き伸ばし]を実行します。

再び[ポリゴン選択(モード)]にしてから、コマンド/[範囲選択]を実行、切り口全体を囲んで選択します。

(FIG 2-88)

ポリゴン面の選択状態をパードビューで確認します。このポリゴンは、連続した引き伸ばし作業には不要です。キーボードの[Delete]キーを押して削除してください。

(FIG 2-89)

もう一度[頂点選択(モード)]にしてから、切り口の頂点グループを[範囲選択]を実行して囲んでいきます。

(FIG 2-90)

コマンド/[選択の引き伸ばし]を実行して、側面図の右方向へドラッグして引き伸ばします。(FIG 2-91)

FIG 2-88

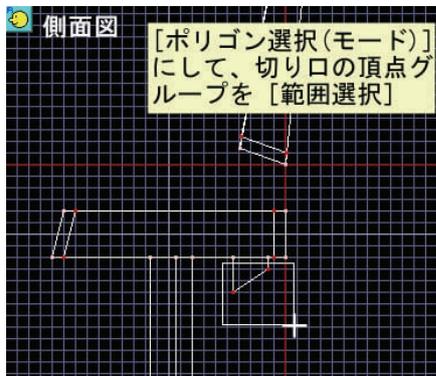


FIG 2-89

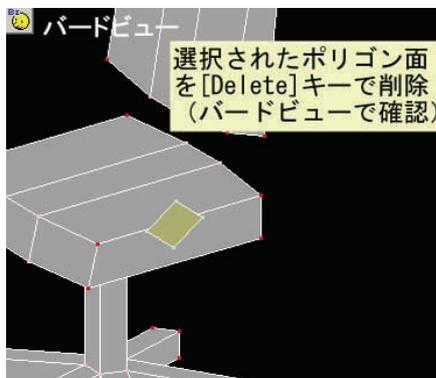


FIG 2-90

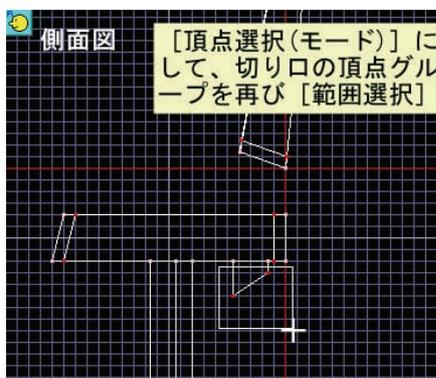
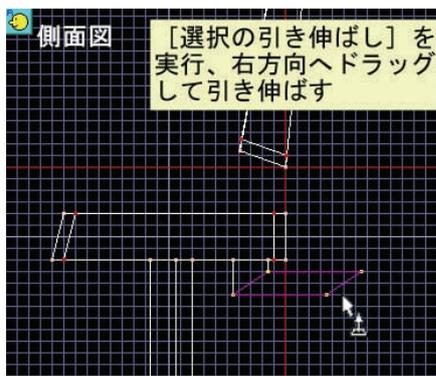


FIG 2-91



トピック

◆ ポリゴン選択(モード)

ショートカットキー
'P'



◆ 範囲選択

ショートカットキー
'B'



◆ 削除

ショートカットキー
'Delete'

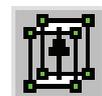
◆ 頂点選択(モード)

ショートカットキー
'V'



◆ 選択の引き伸ばし

ショートカットキー
'E'



トピック

◆ 範囲選択

ショートカットキー
‘B’



◆ 範囲選択

ショートカットキー
‘B’



◆ 選択の引き伸ばし

ショートカットキー
‘E’



◆ バードビューのターン

ショートカットキー
‘T’



(解除は右クリック)

◆ ポリゴン選択(モード)

ショートカットキー
‘P’



[範囲選択]を使用して引き伸ばされた切り口の上側頂点グループを囲んで選択します。(FIG 2-92)

背もたれの方に引き伸ばす為、45度傾くようにドラッグで移動調整します。(FIG 2-93)

FIG 2-92

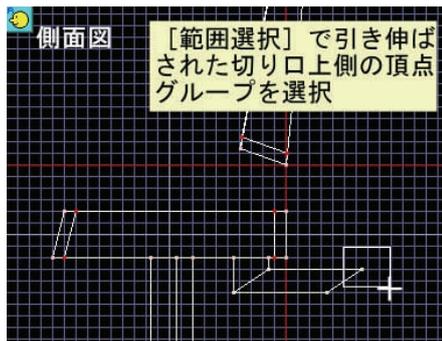
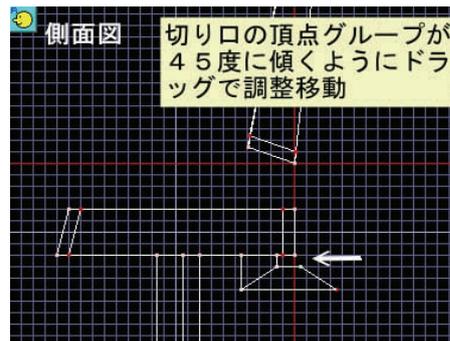


FIG 2-93



角度調整が終了したら、切り口の頂点グループを再び [範囲選択] で囲んで選択します。(FIG 2-94)

コマンド/[選択の引き伸ばし]を実行、背もたれの背面と接触する位置まで上方向へドラッグして引き伸ばします。

(FIG 2-95)

FIG 2-94

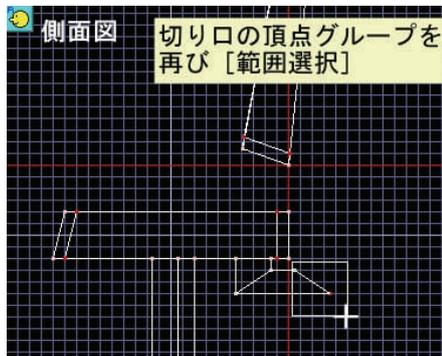
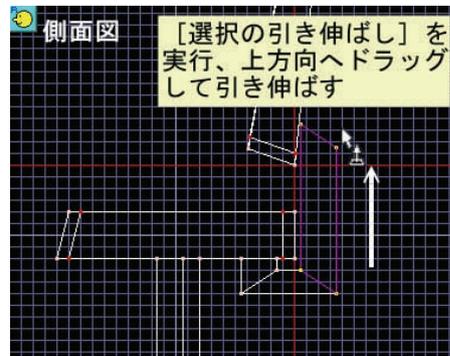
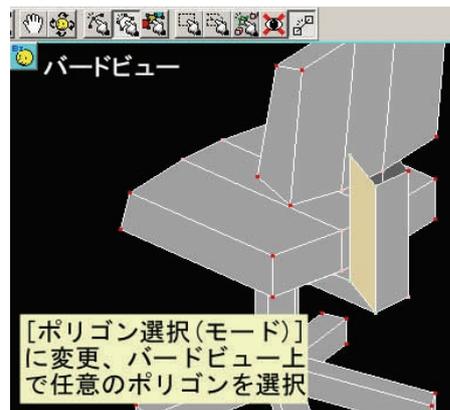


FIG 2-95



引き伸ばされた支柱をバードビュー上で確認します。アングルは [バードビューのターン] で調整してください。続いてモードを [ポリゴン選択] に変更してから、支柱を構成する任意のポリゴン1つを選択します。(FIG 2-96)

FIG 2-96

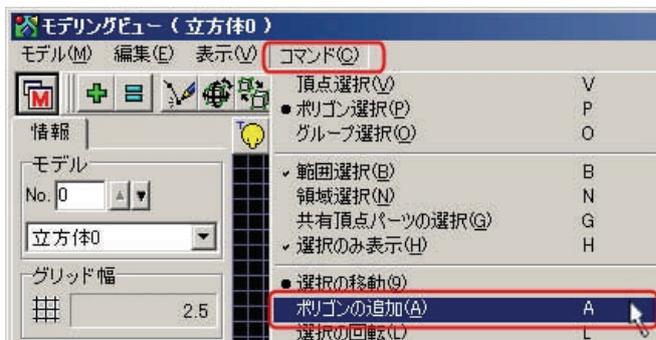


コマンド / [共有頂点パーツの選択] を実行すると、支柱全てが選択状態になります。(FIG 2-97)

この状態でコマンド / [選択のみ表示] を実行、支柱のみ表示させます。表示 [拡大表示] 及び、[視点のスライド移動] を使用して、支柱上部の頂点グループの表示を調整します。(FIG 2-98)

今回のチュートリアルでは、この部分にポリゴン面を追加して支柱の端とすることにします。コマンド / [ポリゴンの追加] を選択してください。(FIG 2-99)

FIG 2-99



既にある頂点に接着するようにポリゴンを新規作成するには、[Shift] キーを併用します。バードビュー上で、[Shift] キーを押した状態で支柱上部の頂点4箇所を反時計回りに順番にクリックしていきます。(FIG 2-100)

FIG 2-97

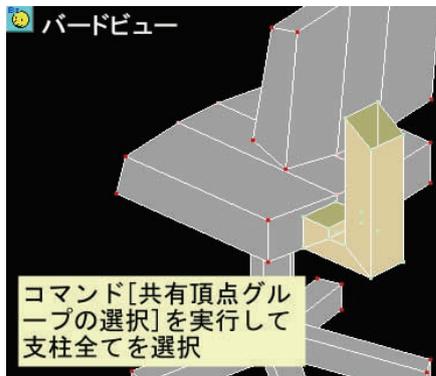
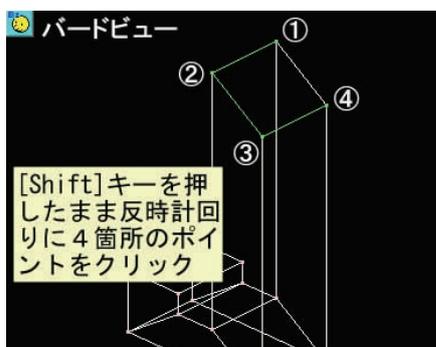


FIG 2-98



FIG 2-100



トピック

◆ 共有頂点パーツの選択

ショートカットキー
'G'



◆ 選択のみ表示

ショートカットキー
'H'



◆ 拡大表示

ショートカットキー
'Shift' + 'I'



◆ 視点のスライド移動

ショートカットキー
'M'



◆ ポリゴンの追加

ショートカットキー
'A'



トピック

◆ 選択のみ表示

ショートカットキー
‘H’



◆ 縮小表示

ショートカットキー
‘Shift’ + ‘O’



◆ 視点のスライド移動

ショートカットキー
‘M’



◆ 頂点選択(モード)

ショートカットキー
‘V’



◆ 範囲選択

ショートカットキー
‘B’



[Shift] キーを離してから [Enter] キーを押してポリゴン面を確定します。(FIG 2-101)

支柱終点のポリゴン面が作成できたら、背もたれに少し食い込むように位置調整を行いたいと思います。

コマンド / [選択のみ表示] を再度選択して、解除します。表示 [縮小表示] 及び [始点のスライド移動] を使用して、側面図上で背もたれと支柱の接点が見えるように表示調整します。(FIG 2-102)

側面図に移り、[頂点選択(モード)] に変更後、[範囲選択] を使用して支柱上部の頂点グループを囲むように選択します。(FIG 2-103)

支柱先端の左側部分が背もたれ背面に少し食い込むように上方向にドラッグ移動します。(FIG 2-104)

FIG 2-101

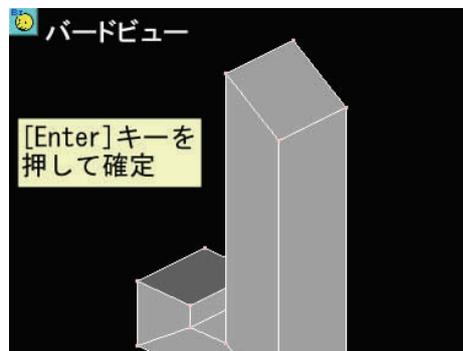


FIG 2-102



FIG 2-103

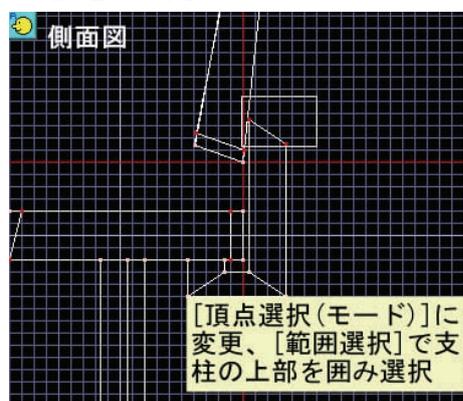
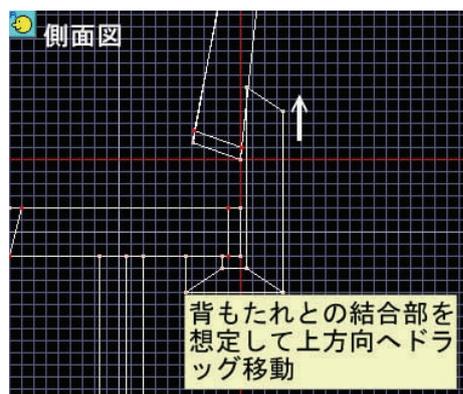


FIG 2-104



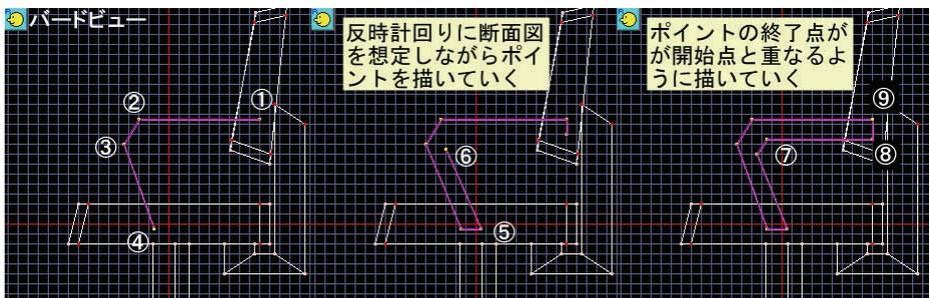
次の編集作業に移る前に一旦 [モデル/モデルの上書き保存] を実行しておきましょう。

オフィスチェアの大部分が完成しましたが、物足りないので肘掛を追加することにします。肘掛は、断面図を描いてから立体化する [掃引体] (そういんたい) という手法で作成します。

コマンド / [掃引体の作成] を選択して、側面図上で肘掛の断面図を想定しながら反時計回りにクリックでポイントを描いていきます。ポイント数の制限はありませんが、後で側面のポリゴンを追加していく関係上、不規則にならないよう注意します。

(FIG 2-105・106)

FIG 2-106



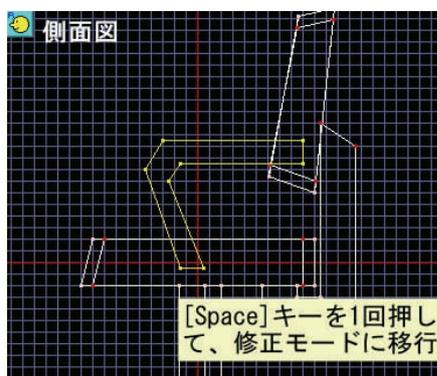
終了点はポイントの開始位置と重なるように描きます。ポイント描画を終えたら、[Space] キーを1回だけ押して修正モードに移行します。修正モードでは描かれたラインが黄色で表示されますが、まだ立体化は行われません。

(FIG 2-107)

FIG 2-105



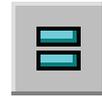
FIG 2-107



トピック

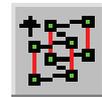
◆ モデルの上書き保存

ショートカットキー
‘Ctrl’ + ‘S’



◆ 掃引体の作成

ショートカットキー
‘F’



トピック

◆ グリッドに合わせる

ショートカットキー
‘Ctrl’ + ‘J’
(解除も同じ)

修正モードでは描画ラインのポイントをドラッグで移動することができます。厚みの統一や、開始点と終了点の位置合わせを行います。なお、表示 / [グリッドに合わせる] 機能が選択されている場合は、グリッド半分ずつの移動になります。微調整を行う場合はこの機能を未選択してください。(FIG 2-108)

もう一度 [Space] キーを押すと、掃引モードに移行します。正面図上で右方向へ肘掛の厚み分だけドラッグしてからマウスボタンを離します。(FIG 2-109)

FIG 2-108

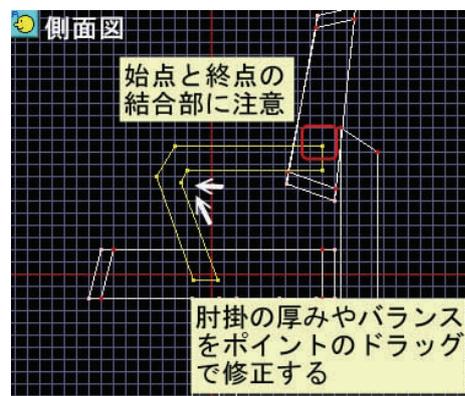


FIG 2-109

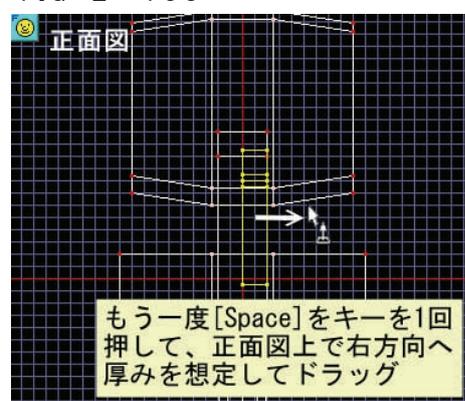
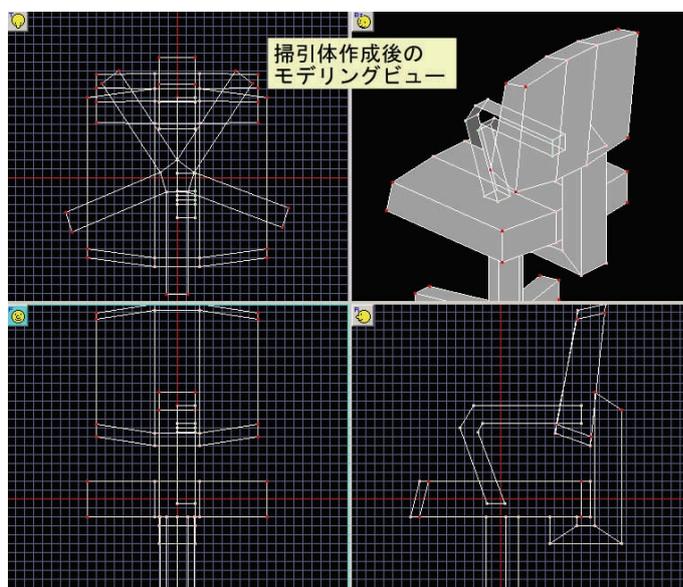


FIG 2-110



掃引体はモデリングビューの中心ライン上に作成されます。この後、側面をポリゴンで埋めていく編集作業と、左右対称に配置する為の移動及び複製作業を行います。(FIG 2-110)

[ポリゴン選択(モード)]に変更後、パードビュー上で肘掛の任意のポリゴン1つを選択します。(FIG 2-111)

コマンド[共有頂点グループの選択]を実行して肘掛全てを選択状態にします。(FIG 2-112)

FIG 2-111

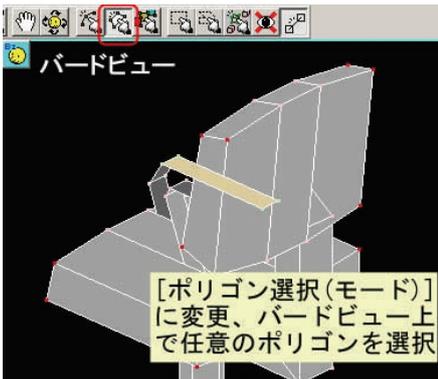
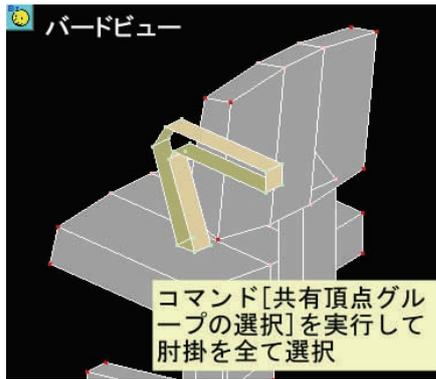


FIG 2-112



正面図上で椅子座面の右方向へドラッグ移動します。(FIG 2-113)

選択状態のまま、コマンド[選択のみ表示]を実行して肘掛のみ表示させます。(FIG 2-114)

FIG 2-113

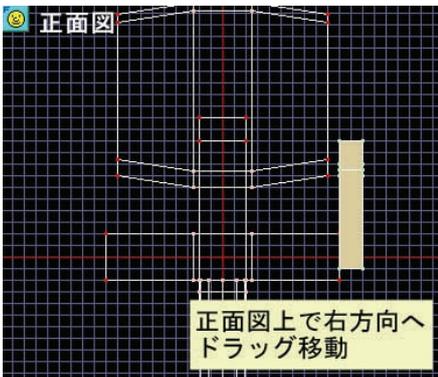
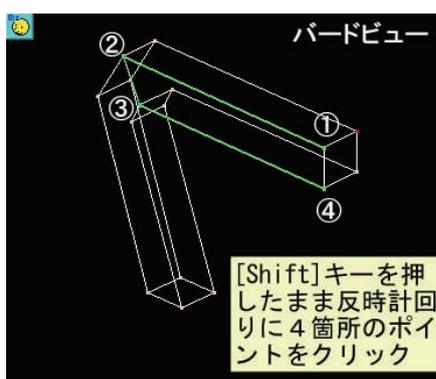


FIG 2-114



パードビュー内の表示を[拡大表示]や[視点のスライド移動]を使用して調整してから、コマンド[ポリゴンの新規作成]を実行します。[Shift]キーを押したまま、反時計回りに4箇所のポイントをクリックします。(FIG 2-115)

FIG 2-115



トピック

◆ ポリゴン選択(モード)

ショートカットキー
'P'



◆ 共有頂点パーツの選択

ショートカットキー
'G'



◆ 選択のみ表示

ショートカットキー
'H'



◆ 拡大表示

ショートカットキー
'Shift' + 'I'



◆ 視点のスライド移動

ショートカットキー
'M'



◆ ポリゴンの追加

ショートカットキー
'A'



トピック

◆ 拡大表示

ショートカットキー
‘Shift’ + ‘I’



◆ 視点のスライド移動

ショートカットキー
‘M’



◆ ポリゴンの追加

ショートカットキー
‘A’



◆ ポリゴンの種類

DigitalLoca3では4角形
ポリゴンと3角形ポリゴン
のみ使用できる。

[Shift] キーを離してから [Enter] キーを押してポリゴン面を確定します。(FIG 2-116)

表示/[拡大表示]・[視点のスライド移動]を使用して、次のポリゴン追加部分の表示を調整します。(FIG 2-117)

FIG 2-116

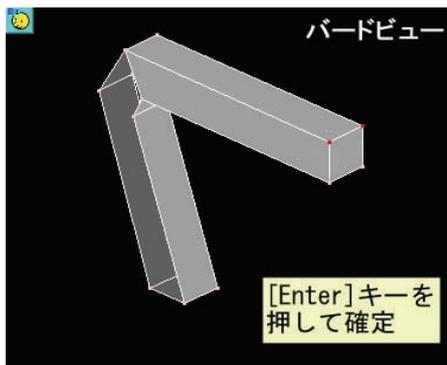
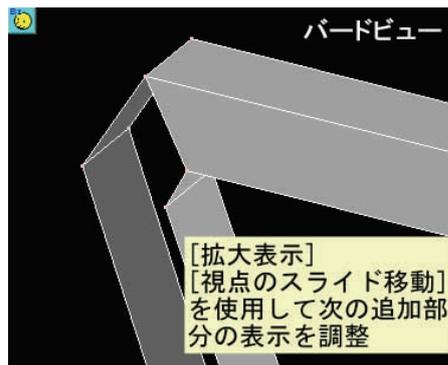


FIG 2-117



同様の手順で、バードビュー上で、[Shift]キーを押した状態で残りの側面にポリゴンを追加していきます。掃引体作成時の描画ポイント数によっては、4つの頂点を持つ4角形ポリゴンが作成できない場合があります。その時は3つの頂点を持つ3角形ポリゴン(3箇所ポイントをクリックして[Enter]キーを押す。)を使用してください。(FIG 2-118~120)

FIG 2-118

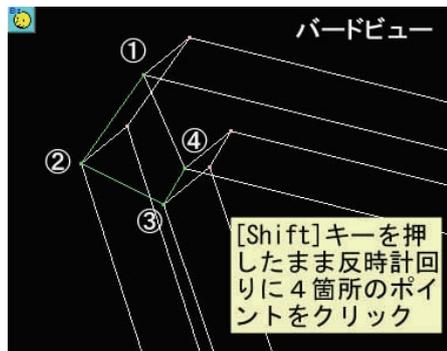


FIG 2-119

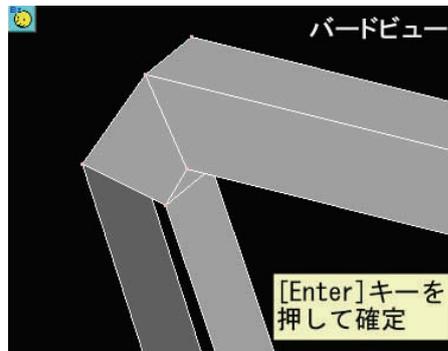
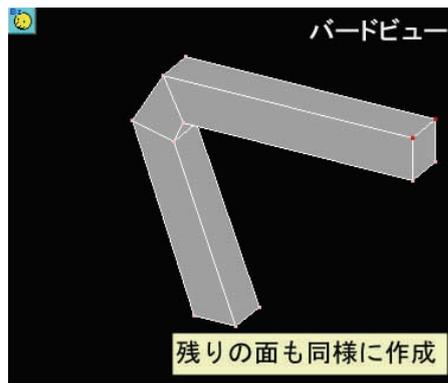
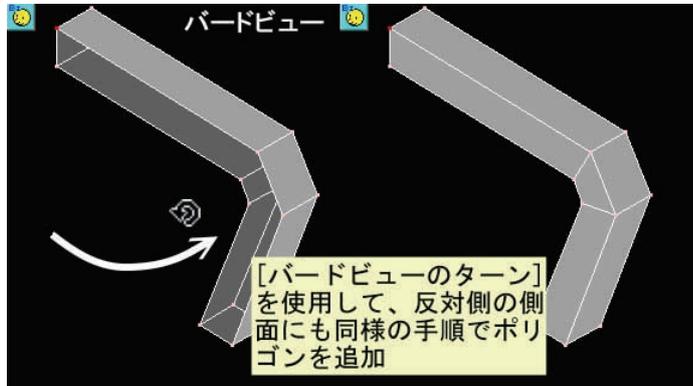


FIG 2-120



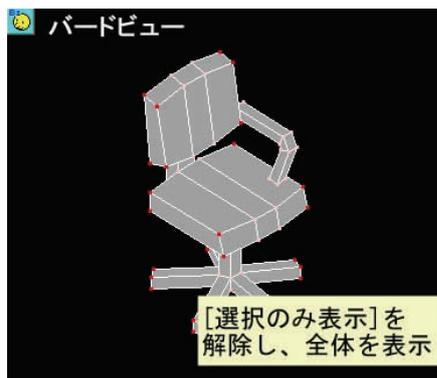
「バードビューのターン」を使用して、肘掛の反対側の側面を表示させます。右クリックで「バードビューのターン」を解除してから、同様にポリゴンを追加していきます。(FIG 2-121)

FIG 2-121



両面ともポリゴンが追加できたら、コマンド「選択のみ表示」を解除して全体を表示させます。(FIG 2-122)

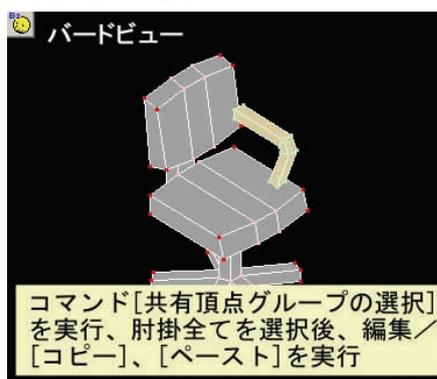
FIG 2-122



◆
3DCGのモデリングには、左右対称のモデルを作成する場合、片側のみ作成し、反対側には元のコピーをスケール反転させる技法がよく使用されます。この技法を使って、もう片方の肘掛を作成してみましょう。

◆
肘掛の任意の部分のポリゴンを1つ選択して、コマンド「共有頂点グループの選択」を実行します。肘掛全てが選択状態になったら、編集／「コピー」を行い、続けて同じ編集メニューから「ペースト」を実行します。(FIG 2-123)

FIG 2-123

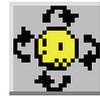


「ペースト」された肘掛は同じ位置に重複して存在しています。この状態のまま、コマンド／「選択の拡大縮小」を実行します。このコマンドを実行

トピック

◆ バードビューのターン

ショートカットキー
‘T’



(解除は右クリック)

◆ 選択のみ表示

ショートカットキー
‘H’



◆ 共有頂点パーツの選択

ショートカットキー
‘G’



◆ コピー

ショートカットキー
‘Ctrl’ + ‘C’

◆ ペースト

ショートカットキー
‘Ctrl’ + ‘V’

◆ 選択の拡大縮小

ショートカットキー
‘S’



トピック

◆ ポリゴンの表裏反転

ショートカットキー
‘R’



◆ バードビューの表示

■ エディット

ショートカットキー
‘Shift’ + ‘1’
頂点と面を結ぶラインをグ
レーで簡易的に塗りつぶ
して表示

■ シェーディング

ショートカットキー
‘Shift’ + ‘2’
陰をともなう立体的なモデ
ル色で表示

■ エディット+シェーディング

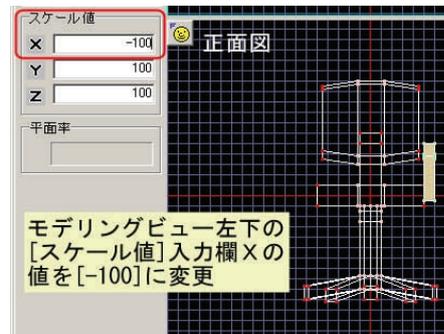
ショートカットキー
‘Shift’ + ‘3’
頂点・面・ラインと、立体的
な陰づけの両方で表示

すると、モデリングビュー左下にある [スケール値] 入力欄がアクティブになります。 Xスケールのみ値を [-100] に変更して [Enter] キーを押します。(FIG 2-124・125)

FIG 2-124



FIG 2-125



スケール反転が確定されると、肘掛のコピーはモデリングビューの中心ラインはさんで対称の位置に移動します。(FIG 2-126)

スケール反転技法は、形状はそのままに位置のみが- (マイナス) 側に移動しますが、スケールにも負の値が与えられる為、ポリゴンの表裏が反転してしまいます。そこで、コマンド / [ポリゴンの表裏反転] を実行、裏になったポリゴン面をクリックして反転させてください。(FIG 2-127)

FIG 2-126

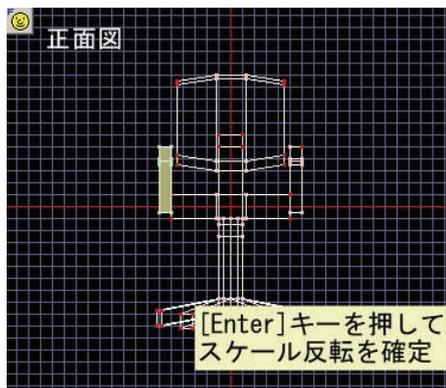
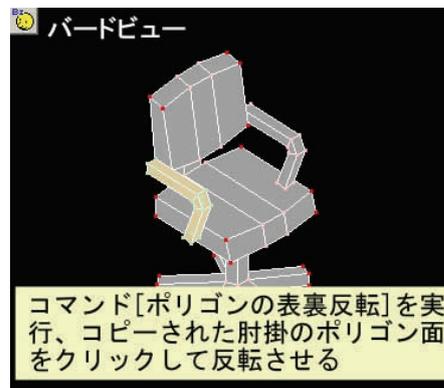
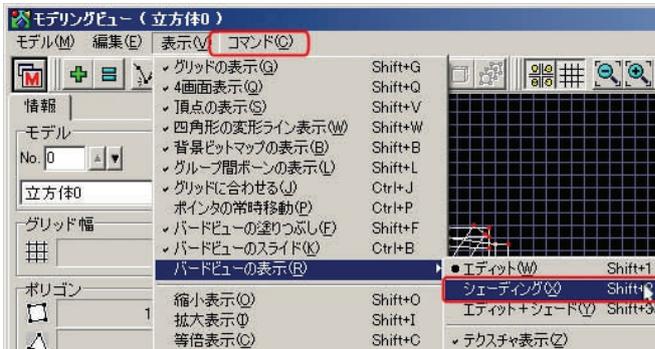


FIG 2-127



最後にキーボードの [Space] キーを押して選択状態を解除してください。これでオフィスチェアの基本的なモデリング編集は終わりですが、モデリングビュー内で3D的な全体像と質感を検証する必要があります。現在のバードビューは、編集作業の為に簡易的な面の塗りつぶし表示になっていますが、このビューのみ色々な表示方法を選択することができます。表示 / [バードビューの表示] から [シェーディング] を選択してください。するとバードビュー内のモデルがシェーディング (陰づけ) された状態で表示されます。(次ページの FIG 2-128・129)

FIG 2-128



立体的にシェーディング(陰づけ)されたバードビュー内のモデルを見ると、面の角度が曖昧な部分が多く見られます。

これは移動や結合を繰り返す編集の為に、ポリゴン面を構成するそれぞれの頂点が確実に結びついていない(=共有されていない)からです。

(FIG 2-129)

FIG 2-129



FIG 2-130



FIG 2-131



頂点を共有させる為に、コマンド / [スムーズ角による頂点の結合] を実行します。

(FIG 2-130)

[UnifyCmdDialog] というダイアログボックスが表示されたら、[モデルのスムーズ角] の数値 [60] を確認して [OK] を押します。(FIG 2-131)

「スムーズ角」についてはトピックに詳しい解説がありますので参照してください。

トピック

■ シェーディング

3D コンピュータ・グラフィックス用語。

「陰づけ」ともいう。モデルの質感を決定する要素で、ポリゴン面をどう描画(塗りつぶす)し、明るさや暗さ(陰)の輝度やそのグラデーション(階調)をどう表現するのかが数学的に定義される。

◆ スムージング角による頂点の結合

ショートカットキー
'I'

■ スムージング角

3D コンピュータ・グラフィックス用語。

スムーズ角は、2つ以上連なるポリゴン面のシェーディング(陰づけ)をどの程度の滑らかさで描画表現するかを決定する。角度が鋭角であれば、直覚的な面と面とのつながりになり、角度が鈍角であれば、滑らかな面のつながりになる。処理の最低条件として、連なる面の各頂点は結合されていなければならない。

トピック

[スムージング角による頂点の結合] 処理が実行されると、各頂点の整合性が計算され、同じ座標にある頂点どうしが共有されて結合します。バードビュー内では頂点とそれを結ぶライン全てが選択状態となって強調表示されます。(FIG 2-132)

[Space] キーを押して選択状態を解除、各ポリゴン面の様子を処理前と比較してみてください。(FIG 2-133)

FIG 2-132



FIG 2-133



[モデルのスムージング角] を [60(度)] に設定したことで、直角的な面のシャープさが鮮明になりました。逆に滑らかにしたい時は、[スムージング角] を「90(度)」以上に設定します。[スムージング角] の調整は、高速描画が必要なゲーム等、あまり多くのデータ量(ポリゴンの数)を処理できない3Dモデルにはかかせない技法です。逆に、映画や商業用CMに使用される高品質CGでは、ポリゴン数を可能な限り増やしつつ、リアルで滑らかな表現を追及する為に、さらに高度なスムージング処理が施されます。

さて、[立方体0] からスタートしてどうにかオフィスチェアらしきものが作成されました。最後にモデル全体を編集/[全てを選択] で全選択し、正面図上で椅子の脚部が赤い中心ラインの上

FIG 2-134

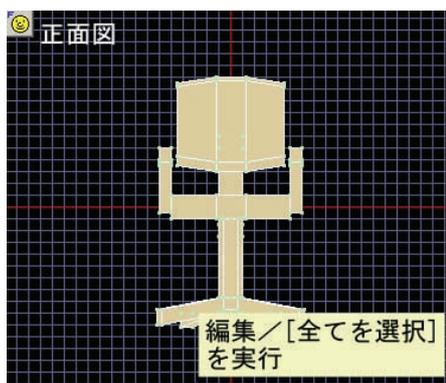
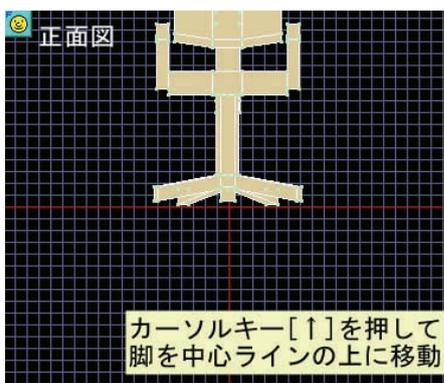


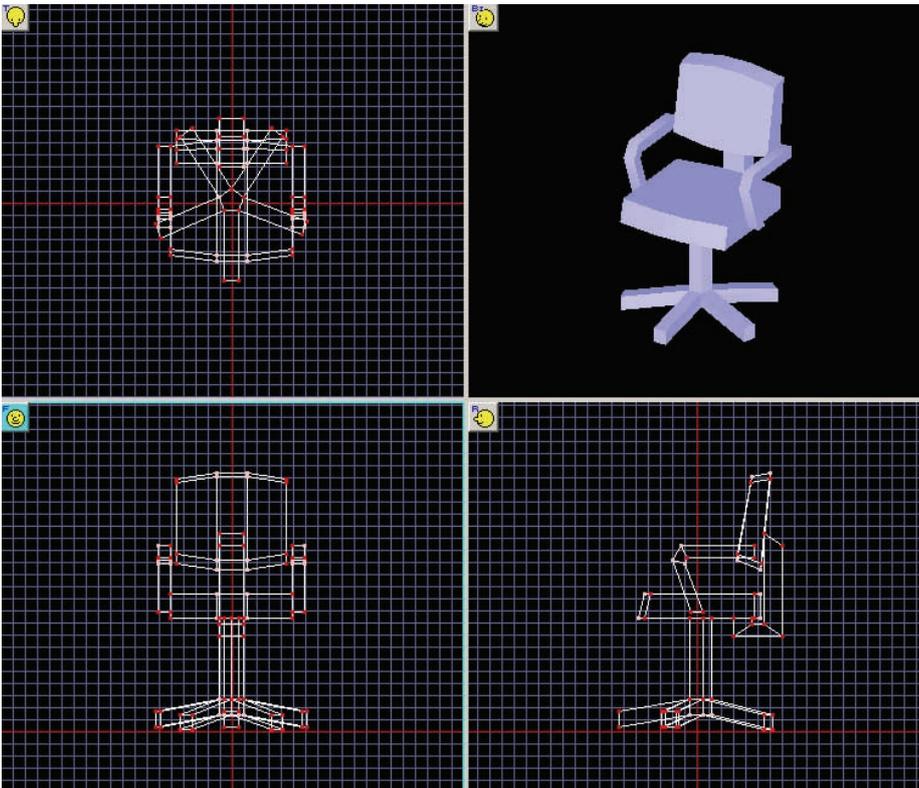
FIG 2-135



乗るようにキーボードの上方向矢印キー [↑] を数回押して移動します。(FIG 2-134・135)

モデル全体の出来具合を確認したら、モデル / [モデルの上書き保存] ('Ctrl' + 'S') を実行してモデリングビューを閉じます。プロジェクト自体の上書き保存も忘れず行ってください。(FIG 2-136)

FIG 2-136



トピック

◆ モデルの上書き保存

ショートカットキー
'Ctrl' + 'S'

