

NJViewer の使い方

1998/03/31

Version 0.73 alpha2

23/03/98 0.73alpha1

- ・ 表示関連を Nindows に変更しました。
- ・ Light Color と Light Direction を操作できるように変更しました。

03/02/98 0.72alpha1

- ・ 複数モデル表示機能を追加しました。
- ・ 背景設定を追加しました。
- ・ 設定値を覚えておくように変更しました。
- ・ 画面表示を bmp ファイルにする機能を追加しました。

01/07/98 0.7alpha4

- ・ Z 軸回転を、シフトキーを押さずに行えるように変更しました。
- ・ モデルで移動後、テクスチャー一覧をしても、モデルが初期の位置に戻らないように変更しました。
- ・ 複数オブジェクトに対応するようにしました。
- ・ 画面表示をファイルに出力する機能を追加しました。

11/07/97 0.7alpha3

- ・ テクスチャー一覧表示機能を追加しました
- ・ 最新の Ninja.lib をリンクしました
- ・ 最新の Nindows.lib をリンクしました
- ・ マウスの回転機能が解除できない不具合を修正しました

11/07/97 0.7alpha2

- ・ バージョン名を統一しました
- ・ テクスチャをロード時にエラーを発生しても読み込めるように変更しました
- ・ 画面初期表示時にモデルサイズ計算処理を最適化するようにしました

10/20/97 1.00alpha1 (PVR)

- ・ PowerVR 版。Seek 機能を追加しました

目次

1.	NJViewer 概要.....	4
2.	動作環境.....	4
3.	インストール.....	4
4.	起動方法.....	4
5.	入力ファイル.....	5
6.	モード	6
6.1	単数または複数モデルのモード	6
6.2	モデル、モーションまたはテクスチャのモード	7
7.	テクスチャー一覧.....	8
8.	シーク機能.....	9
9.	Nindows について.....	11
9.1	Nindows 概要	11
9.2	Nindows の特徴	11
9.3	メニュー	12
9.4	Nindows ユーティリティ	13
9.5	フォントの変更	17
9.6	User メニュー	18
10.	操作.....	22
10.1	キーボード	22
10.2	マウス	23
11.	制限事項.....	23
旧 VIEW 関数使用時の注意.....		24
注意事項 1) の詳細.....		24
注意事項 2) の詳細.....		24
正しい旧 VIEW 関数の使用例.....		25
誤った旧 VIEW 関数の使用例 1		25
誤った旧 VIEW 関数の使用例 2		25
VIEW 構造体		25

1. NJViewer 概要

SoftImage、LightWave、3DStudioMax 等で作成したデータを、コンバータで Ninja 用のモデルとモーションのファイル(Nja、Nam、Nas)に変換します。これらのファイルを用いて、Ninja 上で実際に表示されるモデルを見るためのツールです。

機能は、モデルの表示、テクスチャの表示、モーション（シェイプ）の動作があります。

2. 動作環境

Windows95 が動作するパソコン、Ninja が対応している 3D 画像ボードを用意してください。

3. インストール

Njviewer.exe を任意のディレクトリにコピーします。

4. 起動方法

Njviewer.exe に PVR、NJA(+NAM+NAS)ファイルまたは NSC ファイルをドラッグアンドドロップします。

または Njviewer.exe をダブルクリックして起動します。

5. 入力ファイル

Njviewer で使用できるファイルはいくつかあります。これらの説明を以下に示します。

(1) PVR ファイル

テクスチャファイルです。起動時に選択すると、テクスチャモードで起動します。テクスチャモードはテクスチャを一覧で表示します。

(2) NJA ファイル

Ninja のモデルのファイルです。アスキーファイルで、内容はモデル構造体になっています。

(3) NAM ファイル

Ninja のモーションファイルです。アスキーファイルで、内容はモーション構造体になっています。このファイルを読み込ませるには、NJA ファイルを、先か同時に読み込ませる必要があります。

(4) NAS ファイル

Ninja のシェイプファイルです。アスキーファイルで、内容はシェイプ構造体になっています。このファイルを読み込ませるには、NJA と NAM ファイルを、先か同時に読み込ませる必要があります。

(5) NSC ファイル

Ninja のシーンファイルです。アスキーファイルで、内容は、パスと NJA、NAM、NAS ファイル名で構成されています。このファイルはユーザ側で簡単に変更できます。

このとき、パスは絶対または相対で指定し、カレントの場合は指定しないでください。

ユーザが NSC を作成し読み込むと、テクスチャのグローバルインデックスが重なってしまう可能性があります。この場合、GIGEN.EXE を使用してテクスチャのグローバルインデックスをシーケンシャルにしてください。

例

//NSC 0.72alpha1 NinjaScene(v 0.1)

PATH: c:¥tmp¥jump

jump2_joint.nja

jump2_joint.nam

jump2_armUR.nja

jump2_armUR.nam

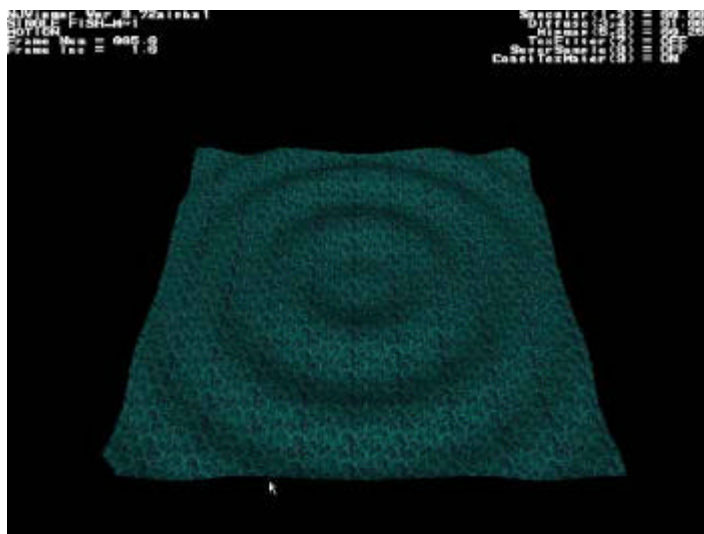
6. モード

6.1 単数または複数モデルのモード

起動時に選択するファイルによって、モードも変わってきます。

(1) SINGLE モード

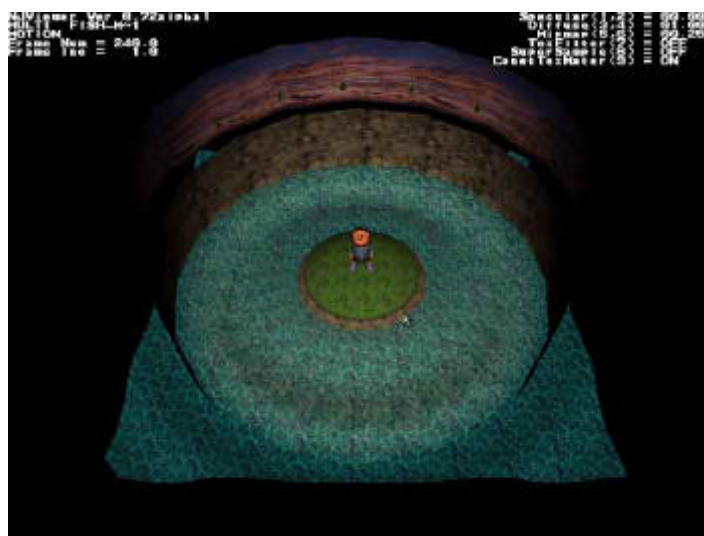
モデルを 1 つずつ表示するモードです。PVR、NJA ファイルを選択するとこのモードで起動します。
F3 キーで、MULTI モードに移行します。



(2) MULTI モード

モデルを複数同時に表示するモードです。NSC ファイルを選択するとこのモードで起動します。F3 キーで、SINGLE モードに移行します。

カレントのモデルは、左上に表示しており(MULTI のとなり)シークや各種操作はこのモデルを対象に動作します。カレントモデルを切り替えたいときは、F11 と F12 KEY 入力してください。



6.2 モデル、モーションまたはテクスチャのモード

起動時に選択するファイルによって、モードも変わってきます。

(1) MODEL ONLY モード

モデルだけを表示するモードで、モーションなどの動作はありません。NJA ファイルを選択すると、このモードで起動します。F4 キー入力で他のモードからこのモードに移行します。

(2) MOTION モード

モデルに加えてモーションを表示するモードです。NJA と NAM ファイルを同時に選択するとこのモードで起動します。また、モーションのデータが読み込まれているときに、F5 または F6 キー入力で他のモードからこのモードに移行します。

また、NAS ファイルを読み込むとシェイプの動作をします。NAS ファイルを読み込むときは、NAM ファイルを読み込んでください。

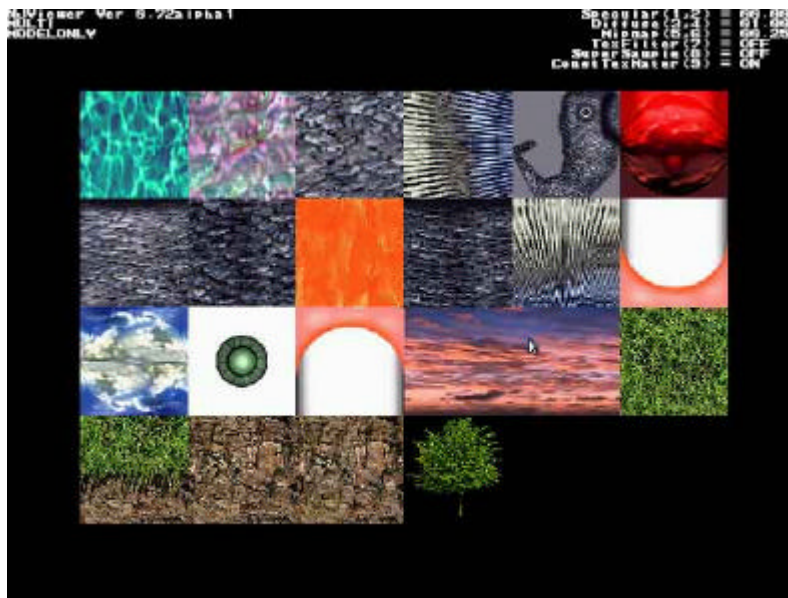


(3) MOTION STEP モード

モーションをステップ表示するモードです。モーションのデータが読み込まれているときに、F7 または F8 キー入力で他のモードからこのモードに移行します。

7. テクスチャ一覧

モデル表示中に D KEY を入力するか、または起動時に PVR ファイルを選択するとテクスチャ一覧を表示します。モデルと同様に、シーク機能で近づいたりできます。

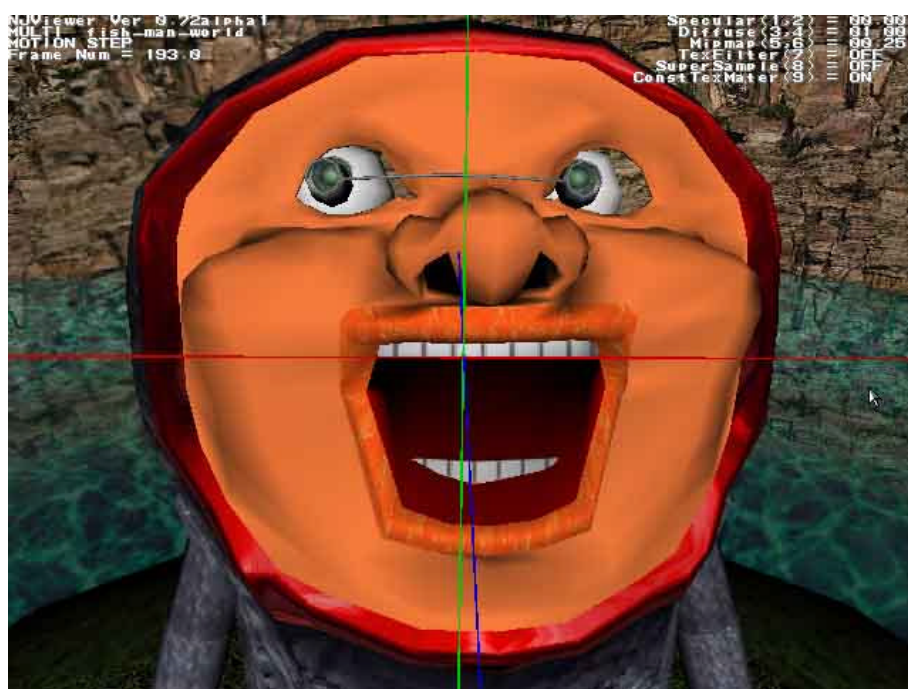
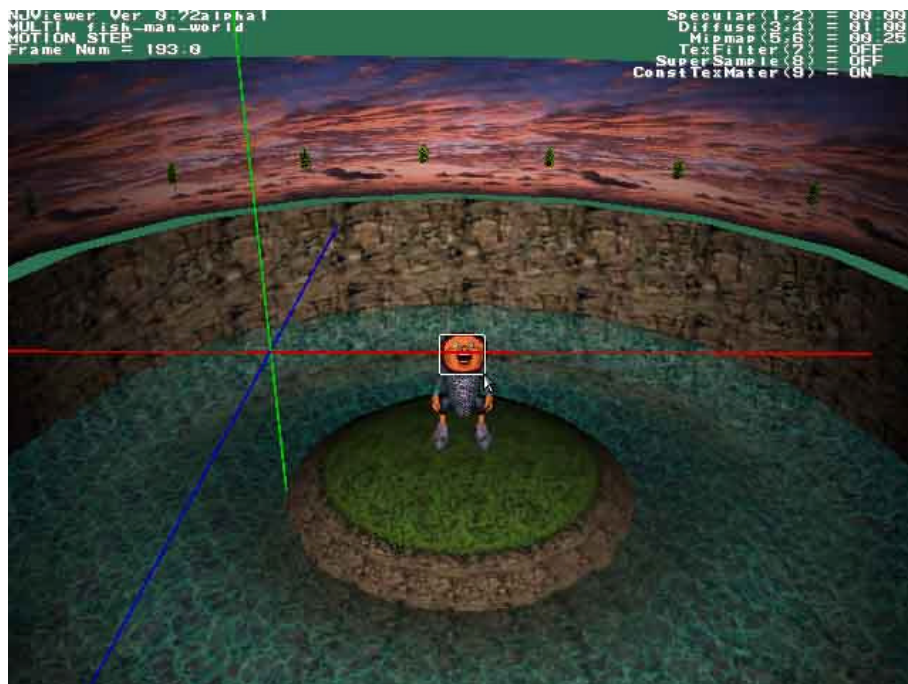


8. シーク機能

この機能を使用すると、表示させたいところを簡単に指定できます。

S キーを押しながら、モデルを左ドラッグすると選択した範囲の大きさが画面表示されるようにビューが近づきます。

シークをしたあと、モデルの中心に回転中心が設定されます。A キーで、軸を表示させるとよく分かります。



右下から左上にドラッグすると逆に遠ざかります。



9. Nindows について

NJViewer には、Nindow という GUI システムが搭載されています。マウスの右クリックでメニューが表示されます。このシステムの説明として、Nindows Tutorial から一部抜粋したものを以下に示します。

Nindows は、ユーザが使用できるものとして、Ninja ライブラリとともに公開されています。**Nindows の詳細は、Nindows Tutorial(Nindows.doc)を参照してください。**

9.1 Nindows 概要

Nindows とは、ゲームを開発する上で欠かせないデバッグやパラメータの調整などを実機およびホストマシン上で行うための、簡易的に使用できる GUI システムです。

9.2 Nindows の特徴

- Windows などの一般的な GUI と同じコントロールが使用できます。
- Nindows API により、アプリケーションで自由にウィンドウを作成することができます。
- 複雑なプログラミングをすることなく、Texture Viewer 等のデバッグに便利なユーティリティが利用できます。
- Nindows で調整したパラメータ等はリアルタイムに確認できるため、ゲームバランス等の調整が迅速に行えます。
- 調整したパラメータを、ホストマシンではファイルに、実機ではバックアップメモリ等に保存できます。(本バージョンではサポートされていません)

9.3 メニュー

- システムメニュー

デスクトップ上でマウスの右ボタンをクリックすると、ポップアップメニューが表示されます。これをシステムメニューと呼び、ここからメニューを選択して Nindows の持つ Nindows ユーティリティを使用します。システムメニューには以下の項目があります。

メニュー項目	説明
Debug	Nindows ユーティリティメニューをポップアップします。
User(Undefined)	ユーザー定義のメニューをポップアップします。Nindows 初期化時には、ユーザーメニューは登録されていないため、淡色表示となり選択できません。
Font	フォントを変更します。
Exit	アプリケーションを終了します。

表 9-1.システムメニューのメニュー項目一覧

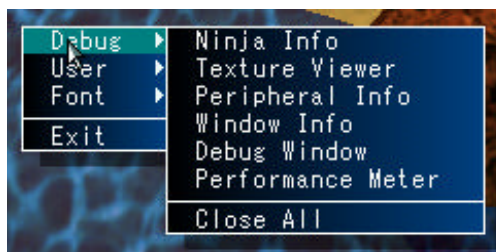


図 9-1.右ボタンでシステムメニューを表示し「Debug」を選択した状態

9.4 Nindows ユーティリティ

システムメニューから「Debug」を選択した時に表示されるメニュー項目が、Nindows の持つ Nindows ユーティリティです。Nindows ユーティリティには以下のものがあります。

名称	説明
Ninja Info	Ninja ライブラリのバージョン番号、その他情報が表示されます。
Texture Viewer	読み込まれているテクスチャをすべて表示することができます。
Peripheral Info	ペリフェラルの情報が表示されます。
Window Info	アクティブなウィンドウの情報が表示されます。
Debug Window	デバッグ文字列を表示するのに便利なウィンドウです。
Performance Meter	アプリケーションの描画パフォーマンスがわかります。

表 9-2.Nindows ユーティティ一覧

9.4.1 Ninja Info ウィンドウ



図 9-2.Ninja Info ウィンドウ

Ninja Info ウィンドウには以下の情報が表示されます。

表示	内容
Ninja Ver.	Ninja ライブラリのバージョン番号
Nindows Ver.	Nindows のバージョン番号
Vertex	投入頂点数
Calc polygon	投入ポリゴン数
Draw polygon	描画ポリゴン数
Texture Memory	テクスチャメモリの空き容量と全容量

表 9-3.Ninja Info ウィンドウの情報

9.4.2 Texture Viewer ウィンドウ

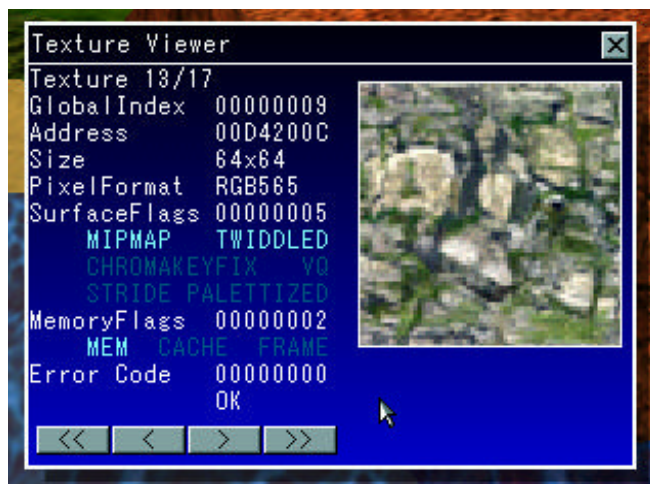


図 9-3.Texture Viewer ウィンドウ

登録されているテクスチャをすべて見ることができます。
4つのボタンでテクスチャを切り替えることができます。
Texture Viewer ウィンドウには以下の情報が表示されます。

表示	内容
Texture	テクスチャ番号と総テクスチャ数
GlobalIndex	グローバルインデックス
Address	テクスチャアドレス
Size	テクスチャサイズ
PixelFormat	ピクセルフォーマット
SurfaceFlags	サーフェースフラグ ハイライト表示されている項目が、そのテクスチャに設定されているフラグです。フラグの詳細はテクスチャ関連ドキュメントを参照してください。
MemoryFlags	メモリフラグ 同じく、そのテクスチャに設定されているフラグです。フラグの詳細はテクスチャ関連ドキュメントを参照してください。
Error Code	テクスチャをロードする際に発生したエラーコードです。ロードに成功した場合は「OK」が表示されます。エラーコードの詳細はテクスチャ関連ドキュメントを参照してください。

表 9-4.Texture Viewer ウィンドウの情報

9.4.3. Peripheral Info ウィンドウ



図 9-4. Peripheral Info ウィンドウ

ペリフェラル（入力装置）の情報を表示するウィンドウです。
「<」、「>」ボタンで表示するペリフェラルポートを選択できます。
Peripheral Info ウィンドウには以下の情報が表示されます。

表示	内容
Port	ペリフェラルポート名
Dev	ポートに接続されているペリフェラル名
ON	押されているボタン情報
OFF	押されていないボタン情報
PRESS	押された瞬間のボタン情報
RELEASE	放された瞬間のボタン情報
X	X 軸値
Y	Y 軸値

表 9-5. Peripheral Info ウィンドウの情報

9.4.4. Window Info ウィンドウ



図 9-5.Window Info ウィンドウ

マウスカーソルの下にあるウィンドウ（アクティブウィンドウ）の情報が表示されます。

Nindows API を使用してアプリケーション独自のウィンドウを作成する場合などのデバッグに使用できます。

Peripheral Info ウィンドウには以下の情報が表示されます。

表示	内容
	アクティブウィンドウのタイトル文字列
x,y	ウィンドウのクライアント領域の左上座標
w,h	ウィンドウのクライアント領域のサイズ

表 9-6. Window Info ウィンドウの情報

x, y 座標は必ずしも画面上の絶対座標ではなく、ウィンドウスタイルに依存します。

9.4.5 デバッグ用ウィンドウ



図 9-6.Debug 用ウィンドウ

このウィンドウには、何も表示されません。

9.4.6 Performace Meter ウィンドウ

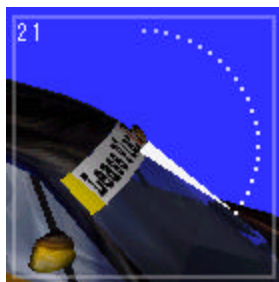


図 9-7.Performance Meter ウィンドウ

アプリケーションのパフォーマンス（計算、描画速度）が直感的にわかります。
メーターが1秒で1周すれば、フレームレートは 60fps です。

9.5 フォントの変更

システムメニューから「Font」を選択することで、通常サイズと大きいサイズのフォントを選択することができます。NTSC モニタに出力した場合など、通常サイズでは文字が読みにくい場合には大きいサイズのフォントを選択してください。

本バージョンでは、フォントサイズ変更に伴うウィンドウのリサイズには対応していません。
フォントを変更する場合、なるべくアプリケーション実行後、他のウィンドウを開く前にフォントを変更するようにしてください。

9.6 User メニュー

システムメニューから「User」を選択した時に表示されるメニュー項目が、NJViewer の持つ NJViewer の GUI 機能です。NJViewer の GUI 機能には以下のものがあります。

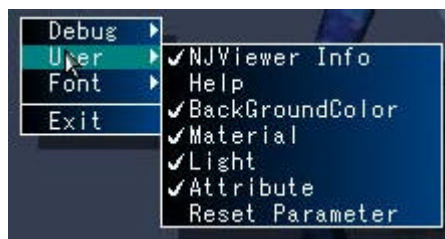


図 9-8. 右ボタンでシステムメニューを表示し「User」を選択した状態

名称	説明
NJViewer Info	NJViewer のバージョン番号、その他情報が表示されます。
Help	ヘルプ表示です。
BackGroundColor	背景色を設定します。
Material	マテリアルの色を設定します。
Light	ライトの各種パラメータを設定します。
Attribute	アトリビュートを設定します。

表 9-7. NJViewer の GUI 機能

9.6.1 NJViewer Info ウィンドウ

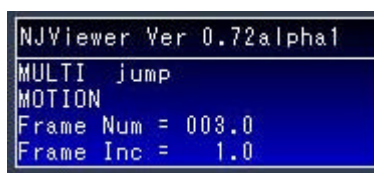


図 9-9. NJViewer Info ウィンドウ

表示	内容
SINGLE or MULTI	SINGLE または MULTI モードの表示をし、その後にファイル名を表示します。
MODELONLY or MOTION or MOTION STEP	モーションの状態が表示されます。
Frame Num	表示フレーム番号が表示されます。
Frame Inc	フレームの増分が表示されます。

表 9-8. NJViewer Info ウィンドウの情報

9.6.2 Help ウィンドウ

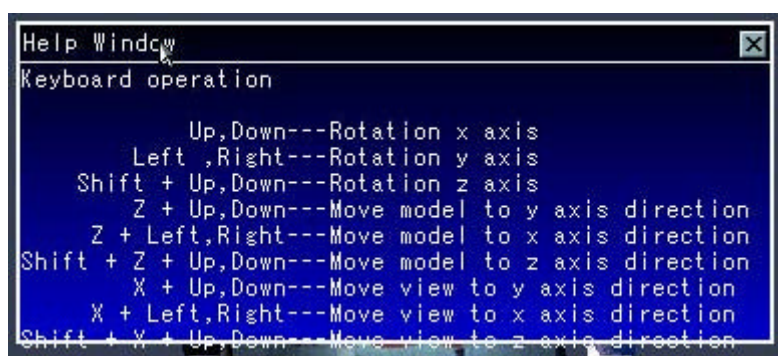


図 9-10. Help ウィンドウ

ヘルプが表示されます。

クライアント領域でマウスを上下に左ドラッグするとスクロールします。

キーボードの PageUp 、 PageDown でページのスクロール、 で 3 行スクロールします。

9.6.3 BackGroundColor ウィンドウ

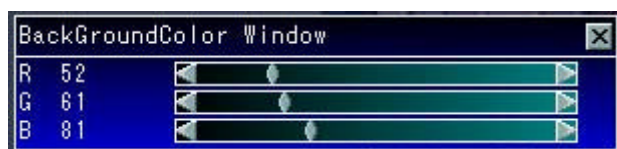


図 9-11. BackGroundColor ウィンドウ

背景色を設定します。スクロールバーで値を設定します。

9.6.4 Material ウィンドウ

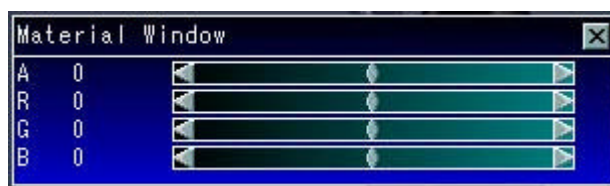


図 9-12. Material ウィンドウ

マテリアルの色を設定します。スクロールバーで値を設定します。

9.6.5 Light ウィンドウ



図 9-13. Light ウィンドウ

スクロールバーで値を設定します。

「<」、「>」ボタンでライトの方向を設定します。

Light ウィンドウには以下の情報が表示されます。

表示	内容
Specular	ライトの Specular 値が表示されます。
Diffuse	ライトの Diffuse 値が表示されます。
Mipmap	Mipmap レベルが表示されます。
Color R	ライトカラー Red 成分が表示されます。
Color G	ライトカラー Green 成分が表示されます。
Color B	ライトカラー Blue 成分が表示されます。
Direction	ライトの方向が表示されます。

表 9-9. Light ウィンドウの情報

9.6.6 Attribute ウィンドウ

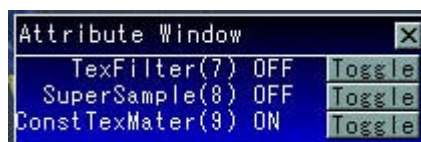


図 9-14. Attribute ウィンドウ

「Toggle」ボタンで状態を切り替えます。

Attribute ウィンドウには以下の情報が表示されます。

表示	内容
TexFilter	テクスチャのフィルターモード ^{注)} を切り替えます。これはテクスチャの拡大時に反映されます。 BILNR でバイリニア補間して表示します。 OFF で補間なしで表示します。
SuperSample	テクスチャの縮小時のスーパーサンプリング ^{注)} を切り替えます。 ON でサンプリング値を平均化して滑らかに表示します。 OFF で平均化しません。
ConstTexMater	テクスチャとマテリアルのカラーの表示 ^{注)} を切り替えます。 ON で、テクスチャのカラーのみを表示します。 OFF で、テクスチャとマテリアルのカラーを合成したものを表示します。

表 9-10. Attribute ウィンドウの情報 注)詳細はスタートアップマニュアルの用語解説を参照してください。

10. 操作

操作とそれに対する動作を示します。

10.1 キーボード

Up,Down	---	モデルが X 軸を中心に回転します。
Left ,Right	---	モデルが Y 軸を中心に回転します。
Shift + Up,Down	---	モデルが Z 軸を中心に回転します。
Z + Up,Down	---	モデルが Y 軸方向に移動します。
Z + Left,Right	---	モデルが X 軸方向に移動します。
Shift + Z + Up,Down	---	モデルが Z 軸方向に移動します。
X + Up,Down	---	ビューが Y 軸方向に移動します。
X + Left,Right	---	ビューが X 軸方向に移動します。
Shift + X + Up,Down	---	ビューが Z 軸方向に移動します。
C + Up	---	モデルがスケールダウンします。
C + Down	---	モデルがスケールアップします。
B	---	マテリアルの Red がダウンします。
G	---	マテリアルの Red がアップします。
N	---	マテリアルの Green がダウンします。
H	---	マテリアルの Green がアップします。
M	---	マテリアルの Blue がダウンします。
J	---	マテリアルの Blue がアップします。
T	---	マテリアルの Alpha がダウンします。
U	---	マテリアルの Alpha がアップします。
Space	---	モデルの位置を初期状態に戻します。
Enter	---	ライトやその他のパラメータを初期状態に戻します。
A	---	座標軸を表示します。
D	---	テクスチャの一覧を表示します。
F1	---	ヘルプを表示します。
F3	---	シングルとマルチモードを切り替えます。
F4	---	モデルのみのモードに変更します。
F5	---	モーションのモードに変更します。押すたびにスピードが遅くなります。
F6	---	モデルのみのモードに変更します。押すたびにスピードが速くなります。
F7	---	逆方向コマ送りモーションのモードに変更します。
F8	---	順方向コマ送りモーションのモードに変更します。
F9	---	モーションファイルを開きます。(11 . 制限事項参照)
F10	---	シェイプファイルを開きます。
F11	---	逆方向にオブジェクトを変更します。
F12	---	順方向にオブジェクトを変更します。
Insert	---	オブジェクトを挿入します。
Delete	---	オブジェクトを削除します。
Print Screen	---	表示画面を bmp ファイルとして出力します。ファイル(Snapshot.bmp) は nja ファイルがあるディレクトリに出力します。
Ctrl	---	同時押しにすると、移動や回転が速くなります。
ALT+F4,ESC	---	終了します。

10.2 マウス

Left drag + Up,Down	---	モデルが X 軸を中心に回転します。
Left drag + Left ,Right	---	モデルが Y 軸を中心に回転します。
Shift + Left drag + Up,Down	---	モデルが Z 軸を中心に回転します。
Z + Left drag + Up,Down	---	モデルが Y 軸方向に移動します。
Z + Left drag + Left,Right	---	モデルが X 軸方向に移動します。
Shift + Z + Left drag+ Up,Down	---	モデルが Z 軸方向に移動します。
X + Left drag + Up,Down	---	ビューが Y 軸方向に移動します。
X + Left drag + Left,Right	---	ビューが X 軸方向に移動します。
Shift + X + Left drag + Up,Down	---	ビューが Z 軸方向に移動します。
C + Left drag + Up	---	モデルがスケールダウンします。
C + Left drag + Down	---	モデルがスケールアップします。
S + Left drag	---	マウスで囲んだ範囲に対してモデルをスケールアップします。

11. 制限事項

- ディスプレイが 1 台の場合は、F9 キーのモーションを開く、F10 キーのモーションを開くまたは Insert キーのオブジェクトを挿入する機能は使用できません。

NJViewer の表示は、PowerVR のフレームバッファに表示しています。これらの機能は、Windows の画面に表示されてしまうため、ディスプレイを 2 台接続しないと使用できません。ディスプレイが 1 台の場合のときモーションを表示する場合は、起動時に NJA と NAM(NAS)ファイルを同時に読み込んでください。また複数のモデルを表示したときには NSC ファイルを読み込んでください。

- モデルの中に 65 頂点以上のポリゴンがある場合、正常に動作しません。

旧 VIEW 関数使用時の注意

njMultiViewMatrix
njRotateViewX
njRotateViewY
njRotateViewZ
njRotateViewXYZ
njTranslateView
njTranslateViewV
njUnitViewMatrix

以上の 8 関数はこれまでの Ninja との互換性を保つために残してある関数です。
将来ライブラリから削除される可能性があるため、新しくプログラムを組む場合には使用しない
ようお願いします。

また、やむなく使用する際には次の注意事項を守るようお願いします。

- 1) njSetView()関数の後で実行する。
- 2) 上記関数の実行後、必ず njClearMatrix()関数を実行する。

以上の 2 点を守らない場合は動作の保証がされません。

注意事項 1) の詳細

旧 VIEW 関数は VIEW 構造体のメンバー、NJS_MATRIX m を直接操作しますが、現在の VIEW では、

Float	px,py,pz;	// 現在の視点の位置
Float	vx,vy,vz;	// 現在の視点の向き (ベクトル)
Angle	roll;	// 現在の視線の Z 軸に対する傾き
Float	apx,apy,apz;	// 基の視点の位置
Float	avx,avy,avz;	// 基の視点の向き (ベクトル)
Angle	aroll;	// 基の視線の Z 軸に対する傾き

の各メンバーを操作し、メンバー、NJS_MATRIX m は njSetView()関数によって更新されます。

従って、njSetView()関数を実行する以前に旧 VIEW 関数を用いて VIEW の操作を行っても、マトリクスは njSetView()関数によって上書きされてしまいます。

注意事項 2) の詳細

VIEW は必ずマトリクススタックに反映されなければなりません。

この機能は現在 njSetView()関数に組み込まれていますが、上記の理由から njSetView()関数を用いて VIEW をマトリクススタックに反映させることはできません。

そのため旧 VIEW 関数を使用する場合、旧 MATRIX 関数、njClearMatrix()の実行を省略することはできません。

正しい旧 VIEW 関数の使用例

```
NJS_VIEW _view_;

njInitView(&_view_);
njSetView(&_view_);
njTranslateView(&_view_, 0.f, 0.f, 1000.f); // etc
njClearMatrix();
```

誤った旧 VIEW 関数の使用例 1

```
NJS_VIEW _view_;

njInitView(&_view_);
njTranslateView(&_view_, 0.f, 0.f, 1000.f); // etc
njSetView(&_view_);
njClearMatrix();
```

njSetView()関数によって VIEW のマトリクスが上書きされてしまうため、VIEW は初期化状態のままになってしまいます。

誤った旧 VIEW 関数の使用例 2

```
NJS_VIEW _view_;

njInitView(&_view_);
njSetView(&_view_);
njTranslateView(&_view_, 0.f, 0.f, 1000.f); // etc
```

njTranslateView()関数を実行した結果がマトリクススタックに反映されていないため、VIEW は初期化状態のままになってしまいます。

VIEW 構造体

```
typedef struct {
    Float   px,py,pz;      // 現在の視点の位置
    Float   vx,vy,vz;      // 現在の視点の向き（ベクトル）
    Angle   roll;          // 現在の視線のZ軸に対する傾き
    Float   apx,apy,apz;   // 基の視点の位置
    Float   avx,avy,avz;   // 基の視点の向き（ベクトル）
    Angle   aroll;         // 基の視線のZ軸に対する傾き
    NJS_MATRIX m;         // ビューマトリクス
} NJS_VIEW;
```