



第2部

Nindows2 ライブラリ編

この章では、Nindows2 についてライブラリ
全般にわたり解説します。

目次

1 Nindows2 ライブラリについて	3
1.1 Nindows2 の概要	3
1.1.1 Nindows2 の特徴	3
2 Nindows2 アプリケーションの作成方法	4
2.1.1 Nindows2 の組み込み	4
3 Nindows2 と Nindows2 ユーティリティ	5
3.1.1 Nindows2 の操作	5
3.1.2 Nindows2 ユーティリティ	6
3.1.3 フォントの変更	12
3.1.4 Nindows2 アプリケーションの終了	12
4 Nindows2 API の使い方について	13
5 関数一覧	14

1 Nindows2 ライブラリについて

ここでは、Nindows2 ライブラリについて基本的な部分の説明をします。

1.1 Nindows2 の概要

Nindows2 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等を、実機及びホストマシン上で行うための、簡易的に使用が可能な GUI システムです。

1.1.1 Nindows2 の特徴

- Windows 等の一般的な GUI と同じコントロールでの使用が可能になります。
- 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。
- Nindows2 で調整したパラメータ等をリアルタイムに確認が行えるため、ゲームバランス等の調整が迅速に行えます。

2 Nindows2 アプリケーションの作成方法

ここでは、既存の Ninja2 アプリケーションに Nindows2 を組み込む方法を解説します。

Nindows2 の組み込みについては、ソースファイルにわずか数行の変更を加えるのみで、簡単に Nindows2 の機能が利用可能になります。

2.1.1 Nindows2 の組み込み

- (1) Ninja2 アプリケーションを準備する
- (2) Nindows2 のヘッダファイルをインクルードする
ソースファイルに、次の 1 行を追加します。

```
#include <Nindows2.h>
```

- (3) Nindows2 の初期化関数をコールする
NjInitTextureEx() の呼び出し後に、次の 1 行を追加します。

```
nwInitialize(NULL);
```

- (4) Nindows2 の実行関数をコールする
UserMain の最後の return USER_CONTINUE を、次のように変更します。

```
Return nwMainExecute();
```

- (5) Nindows2 の終了関数をコールする
njExitSystem() の呼び出し前に、次の 1 行を追加します。

```
nwFinalize();
```

- (6) Nindows2 のライブラリをリンクする
プロジェクトに Nindows2.lib を加えます。

- (7) Nindows2 ライブラリの組み込みを終了する

以上の作業により、Nindows2 ライブラリの組み込みが終了します。

その結果、Nindows2 の Nindows2 ユーティリティの使用及び Nindows API 関数の呼び出しが可能になります。

3 Nindows2 と Nindows2 ユーティリティ

ここでは、Nindows2 の操作方法や、Nindows2 ユーティリティの解説をします。

3.1.1 Nindows2 の操作

Nindows2 を組み込んだアプリケーション（以下、Nindows2 アプリ）を実行すると、画面上にマウスカーソルが表示され、画面上をマウスの操作に応じて動かすことが可能になります。

- システムメニュー

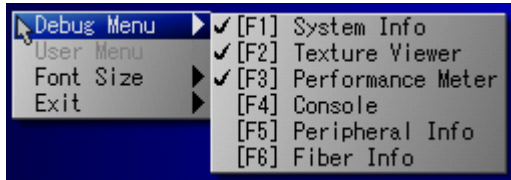
デスクトップ上でマウスの右ボタンをクリックすると、ポップアップメニューが表示されます。このメニューをシステムメニューと呼びます。

システムメニューには、以下の項目があります。

- システムメニューのメニュー項目一覧

メニュー項目	説 明
Debug Menu	Nindows2 ユーティリティメニューをポップアップします。
User Menu	ユーザー定義のメニューをポップアップします。 Nindows2 の初期化時には、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 Size	画面表示フォントの大きさを変更します。
Exit	Nindows2 アプリケーションを終了し、「Main Loop」か「Boot Rom」の画面へ移行します。

- 右ボタンでシステムメニューを表示し、「Debug Menu」を選択した状態の画面



「Debug Menu」を選択した場合に、Nindows2 が持つ Nindows2 ユーティリティのメニューがポップアップし、表示されます。その際に、項目を選択し実行することが可能です。

ノート デスクトップ上でファンクションキーを直接押すことにより、「Debug Menu」を表示せずに該当項目を実行することが可能です。

3.1.2 Nindows2 ユーティリティ

システムメニューから「Debug Menu」を選択した時に表示されるメニュー項目が、Nindows2 の持つ Nindows2 ユーティリティです。

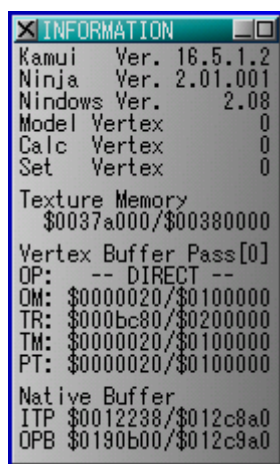
Nindows2 ユーティリティには、以下の項目があります。

- Nindows2 ユーティティ一覧

名 称	説 明
[F1] System Information	システムのバージョン番号、その他関連情報が表示されます。
[F2] Texture Viewer	読み込まれているテクスチャを、全て表示することができます。
[F3] Performance Meter	アプリケーションの描画パフォーマンスを確認できます。
[F4] Debug Console	デバッグ文字列を表示するのに便利なウィンドウです。
[F5] Peripheral Information	ペリフェラルの詳細な情報が表示されます。
[F6] Fiber Information	ファイバの詳細な情報が表示されます。

名称の[F1]～[F6]は、キーボードを接続した際のショートカットキーです。

(1) System Information ウィンドウ



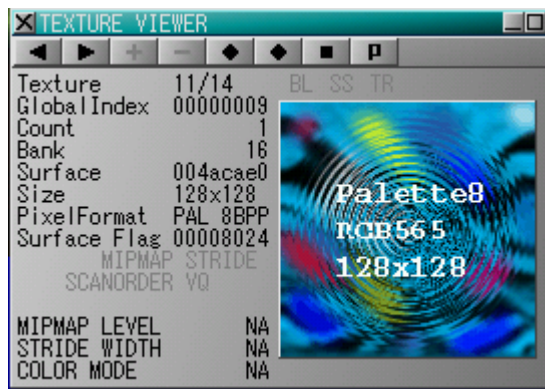
- System Information ウィンドウ

System Information ウィンドウには、以下の情報が表示されます。



● System Information ウィンドウの情報

表 示	内 容
Kamui Ver.	Kamui2 (ライブラリ) のバージョン番号です。
Ninja Ver.	Ninja2 (ライブラリ) のバージョン番号です。
Nindows Ver.	Nindows2 (ライブラリ) のバージョン番号です。
Model Vertex	DrawModel 等の描画関数で投入された頂点数です。
Calc Vertex	DrawModel 等の描画関数で計算された頂点数です。
Set Vertex	DrawModel 等の描画関数でレンダラに渡された頂点数です。
Texture Memory	テクスチャメモリの空き容量と全容量です。
Verftex Buffer Pass [0]	バーテックスバッファの使用容量 (過去の最大値)、最大値 -- NO USE --は、そのリストタイプの使用不可を示します。 -- DIRECT --は、そのリストタイプの直接転送を示します。
Native Buffer	内部で使用する描画用バッファ (VRAM) の最大容量と、現在の使用量、最大使用量です。使用量が最大容量をオーバーするような場合は、描画を減らして下さい。








(2) Texture Viewer ウィンドウ



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事が可能です。その際に、「」「」ボタンでテクスチャを切り替えることができます。

● Texture Viewer ウィンドウのアイコンの情報

表 示	内 容
	テクスチャ表示時は番号の切り替え、パレット表示時はバンクの切り替えです。
	テクスチャ表示時は番号の切り替え、パレット表示時はバンクの切り替えです。
	ミップマップありの時に、ミップマップレベルを切り替えます。
	バイリニアフィルタの ON/OFF (ON の時はメニューの BL が反転します) です。
	スーパーサンプリングフィルタの ON/OFF (ON の時はメニューの SS が反転します) です。
	テクスチャの ON/OFF (ON の時はメニューの TR が反転します) です。
	表示モードの切り替え (パレット / テクスチャ) を行います。 パレット表示モードは、BANK (0、16、32、48) を LR ボタンで切り替えます。

Texture Viewer ウィンドウには、以下の情報が表示されます。

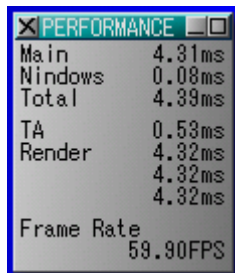
● Texture Viewer ウィンドウの情報

表 示	内 容
Texture	表示中のテクスチャの番号と現在登録されているテクスチャの枚数を表示します。
GlobalIndex	表示中のテクスチャのグローバルインデックスです。
Count	表示中のテクスチャのリファレンスカウンタです。
Bank	表示中のテクスチャのパレットバンクです。
Surface	表示中のテクスチャの VRAM 内のオフセットです。
Size	表示中のテクスチャのサイズです。
PixelFormat	表示中のテクスチャのピクセルフォーマットです。
Surface Flag	表示中のテクスチャのフォーマットです。 MIPMAP : ミップマップあり STRIDE : ストライド SCANORDER : スキャンオーダー (NO TWIDDLED) VQ : VQ
MIPMAP LEVEL	ミップマップありの場合 : 表示しているミップマップのレベルです。
STRIDGE WIDTH	ストライドの場合 : ストライドレジスタの値です。
COLOR MODE	パレット表示モード : カラーモードを表示します。

● テクスチャのスクロールと拡大縮小

ウィンドウ内で A ボタンを押せばなしの状態にすると、カーソルがロックされテクスチャ操作モードになります。ここで十字ボタンを使用して、テクスチャのスクロール及び LR ボタンで拡大縮小ができます。

(3) Performance Meter ウィンドウ



PERFORMANCE	
Main	4.31ms
Nindows	0.08ms
Total	4.39ms
TA	0.53ms
Render	4.32ms
	4.32ms
	4.32ms
Frame Rate	59.90FPS

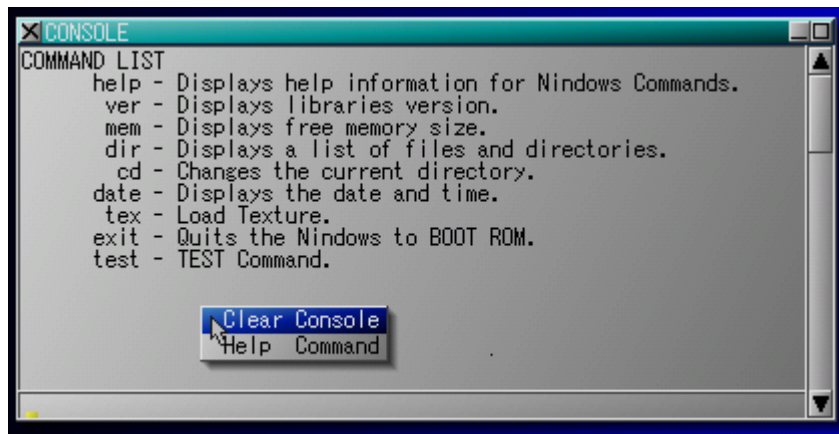
● Performance Meter ウィンドウ

アプリケーションのパフォーマンスを表示します。

● Performance Meter ウィンドウの情報

表 示	内 容
Main	Total タイムから Nindows タイムを引いた値を表示します。
Nindows	nwMainExecute に要した時間を表示します。
Total	njStartDraw から、njRender までの時間を表示します。
TA	頂点リストの DMA 転送に掛かった時間を表示します。
Render	レンダリングの処理に掛かった時間 (過去 3 回分) を表示します。
Frame Rate	nwMainExecute が呼ばれる間隔から計算された値を表示します。

(4) Debug Console ウィンドウ



- Debug Console ウィンドウ

デバッグ情報の表示とコマンドラインからのデバッグコマンドを実行できます。デバッグコマンドを入力する際には、必ずキーボードを接続して下さい。ただしその際に、キーボードが使用可能でなければなりません。

注 意 このデバッグコマンドを使用した場合、アプリケーションの動作を保証できませんので注意して下さい。

- メニューのクリア及びヘルプコマンドの表示

Debug Console ウィンドウ上で、マウスの右ボタンをクリックすると「Clear Console」「Help Command」を選択するメニューが表示されます。

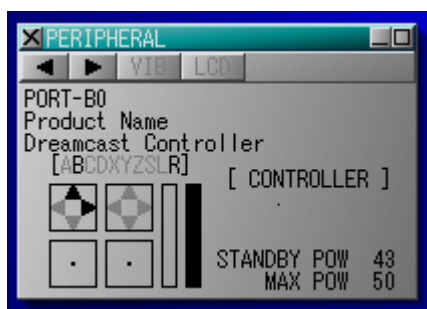
「Clear Console」：Debug Console ウィンドウの表示内容をクリアします。

「Help Command」：コマンドリストの一覧とコマンドの簡単な動作説明の表示を行います。

- Debug Console ウィンドウの情報

入力命令	内容
help	コマンドリストの一覧とコマンドの簡単な動作説明の表示を行います。
ver	メインメモリに存在するライブラリのバージョン情報です。 メモリ上のゴミにも反応してしまいます。その際には、アドレスからユーザープログラムに含まれるライブラリかを判断して下さい。
mem	メインの残りヒープ容量と Nindows の残りヒープ容量を表示します。
dir	カレントディレクトリのリストを表示します。
cd	カレントディレクトリを変更します。
date	RTC?の日付と時間を表示します。
tex	GD からテクスチャを読み込みます。引数にファイル名を指定して下さい。
exit	BOOT ROM へ戻ります。

(5) Peripheral Information ウィンドウ



● Peripheral Information ウィンドウ

ペリフェラルの情報を表示するウィンドウです。

「」「」ボタンで、メニューに表示するコントロールポートを選択できます。

「VIB」：「ぶるぶるぱっく」を選択した時に反転します。

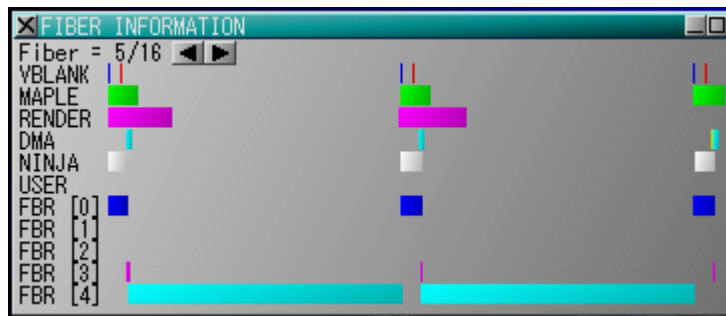
「LCD」：Visual Memory 等の LCD 液晶画面を装備したペリフェラルを選択した時に反転します。対応しているペリフェラルが挿入されていない場合は、それぞれ淡色表示になります。

Peripheral Information ウィンドウには、以下の情報が表示されます。

● Peripheral Information ウィンドウの情報

表 示	内 容
PORT	選択中のコントロールポート名を表示します。
Product Name	ペリフェラルの製品名が表示されます。
[ABCDXYZSLR]	押されているボタンの情報が表示されます。
[CONTROLLER]	コントロールデバイスの表示 [CONTROLLER] [コントローラ (パッド等)] [STORAGE] [ストレージ (ビジュアルメモリ等)] [LCD] [液晶 (ビジュアルメモリ等)] [TIMER] [タイマ (ビジュアルメモリ等)] [SOUNDINPUT] [音声入力 (マイク等)] [KEYBOARD] [キーボード] [LIGHTGUN] [ライトガン (ガンコン等)] [VIBRATION] [振動デバイス (ぶるぶるぱっく等)] [POINTING] [ポインティングデバイス (マウス等)] [CAMERA] [カメラ]
STANDBY POW	スタンバイ時に消費する電流を表示します。
MAX POW	最大消費電流を表示します。

(6) Fiber Information ウィンドウ



● Fiber Information ウィンドウ

Fiber Information ウィンドウには、以下の情報が表示されます。

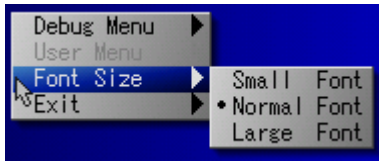
● Fiber Information ウィンドウの情報

表 示	内 容
Fiber	使用中のファイバの数と最大数です。 デフォルトでメインファイバが存在するので、1 以上の数値になります。
VBLANK	青いラインが IN、赤いラインが OUT を示します。通常、IN と IN の間隔は 16.67ms になります。
MAPLE	MAPLE DMA が動作している時間を示します。
RENDER	レンダラが動作している時間を示します。
DMA	黄色いラインは CH2DMA の終了、水色は頂点転送中の時間を示します。
NINJA	njStartDraw から njRender までの時間を示します。
USER	nwUserInfo が呼ばれたタイミングを示します。
FBR [n]	各ファイバが動作している時間です。ただし、ファイバの切り替えを一切行わなかった場合は、表示されません。

ファイバのデフォルト設定は 0 番がメインファイバ、1 ~ 4 番は syMng が使用しています。ユーザーが Create したファイバは 5 番以降になります。

3.1.3 フォントの変更

- 右ボタンでシステムメニューを表示し、「Font Size」を選択した状態の画面



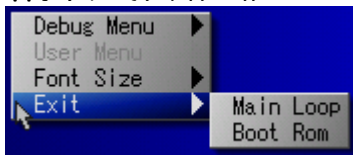
システムメニューから「Font Size」を選択することで、小さいサイズ（Small Font）と通常サイズ（Normal Font）と大きいサイズ（Large Font）の3種類のフォント表示のモードを選択することが可能になります（デフォルトの設定は Normal Size です）。

NTSC モニタに出力した場合等で、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下さい。

注 意 フォントサイズの変更に伴う、ウィンドウのサイズの変更には対応していませんので、フォントを変更する場合は、なるべくアプリケーション実行後に行い、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下さい。

3.1.4 Nindows2 アプリケーションの終了

- 右ボタンでシステムメニューを表示し、「Exit」を選択した状態の画面



「Exit」を選択した場合に、Nindows2 のアプリケーションを終了し「Main Loop」「Boot Rom」の画面へ移行するメニューがポップアップし、表示されます。

「Main Loop」：Nindows2 ユーティリティを終了し、メインループ画面へ移行します。

「Boot Rom」：Nindows2 ユーティリティを終了し、ブートルーム画面へ移行します。

4 Nindows2 API の使い方について

Nindows2 API の使い方については、SEGA Library Version 2.0 の Sample の NindowsSDK 中のサンプル (Sample00、Sample01、Sample02) のそれぞれのソースファイルのコメントを参考に作成して下さい。

- サンプルの内容

フォルダ名	内 容
Sample00	ウィンドウ作成の基本について
Sample01	Nindows を使ったメニューやフォント等の作り方について
Sample02	ボタンとビットマップの作り方について

5 関数一覧

Nindows2 ライブラリでは、以下の関数がサポートされています。
詳しい内容については、関数リファレンスをご参照下さい。

関数名	機 能
nwInitialize	Nindows2 ライブラリの初期化
nwFinalize	Nindows2 ライブラリの終了
nwGetWorkSize	Nindows2 で必要なワークサイズの取得
nwMainExecute	Nindows2 の実行
nwCalcExecute	Nindows2 の計算部分の実行
nwDrawExecute	Nindows2 の描画部分の実行
nwExecuteCommand	登録したコマンドの実行
nwDbPrint	デバッグ文字列の出力
nwDbPrintf	デバッグ文字列の出力（書式付）
nwRegisterCommand	Nindows2 へのコマンド登録
nwUnregisterCommand	登録したコマンドの開放
nwUserInfo	Fiber Info ウィンドウの USER エリア出力