

M.U.G.E.N. SFF ファイル デコーダ 説明書

なにをするソフトなのか？

既存のSFFファイルを解析し、再構築に必要な画像データ、及びその登録情報を出力します。その際に、画像データの物理的な並び順を変更し、WinMugenにおける色化けを回避します。

具体的には、次の内容を実行します。

- ・ SFFに登録されている全PCXデータを、一定の法則にて名前をつけて保存する
- ・ 上記PCXを用いて、sprmakerを使ってSFFを作成するためのパラメータテキストを出力する
- ・ 上記パラメータを用いてsprmakerを実行するためのバッチファイルを出力する
出力されたバッチファイルを実行することで、SFFファイルが作成されます。

出力されるファイル名の命名方法は、次の通りです。

- ・ SFFファイル名から拡張子を除いたものをサフィックスとする
- ・ PCXの場合、「サフィックス_グループ番号_アイテム番号.PCX」となる
- ・ パラメータテキストの場合、「サフィックス_SFF.txt」となる
- ・ バッチファイルの場合、「サフィックス_SFFBuild.bat」となる
(厳密には、「パラメータテキスト名(拡張子除く) + Build.bat」)

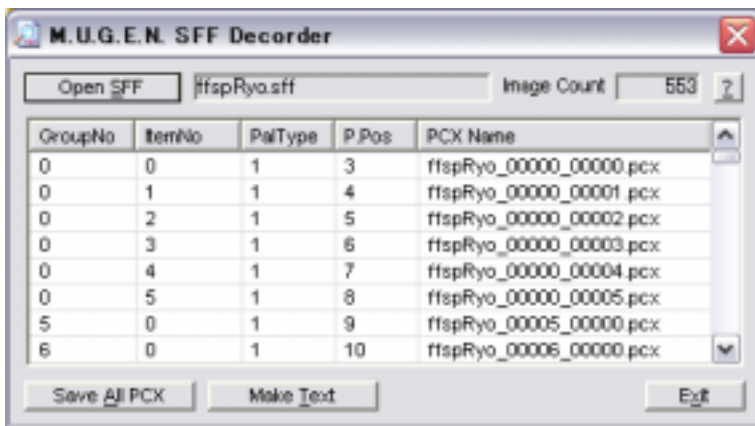
例: SFFファイル名が「ffspRyo.sff」の場合

PCXファイル名 ffspRyo_00100_00000.pcx (グループ番号100、アイテム番号0の場合)

パラメータテキスト ffspRyo_SFF.txt

バッチファイル ffspRyo_SFFBuild.bat

画面構成 及び 使い方



このボタンを押下、SFFファイルを読み込みます。

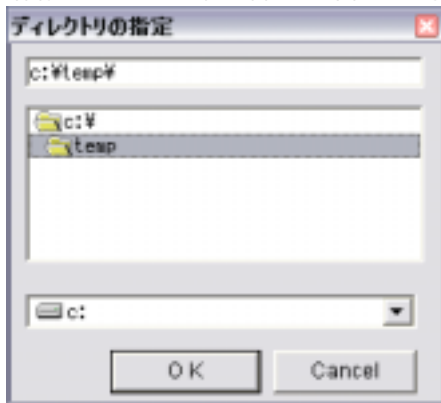
SFFを読み込むと、ここに構成情報が表示されます。

なお、表示されるデータは、WinMugen で色化けを回避するための順に、自動的にソートされます。表示される項目の意味は次の通りです。

- GroupNo グループ番号
- ItemNo アイテム番号
- PalType パレットタイプ(1:共有パレット / 0:独自パレット)
- P.Pos SFF内における物理的な格納順
- PCX Name PCX出力時のファイル名(自動的に命名されます)

このボタンを押下、PCXファイルを全て保存します。

保存先ディレクトリを問い合わせる画面が表示されますので、保存先を指定して下さい。



このボタンを押下、SprMaker用のパラメータテキスト 及び バッチファイルを出力します。コマンドダイアログが表示されますので、保存先とテキストファイル名を指定します。

保存先は、PCXファイルの保存先と同じにして下さい。

変更することも出来ますが、その場合はバッチファイルが正しく動作しなくなります。

SffDecoderを終了します。

バージョン情報を表示します。

技術的なお話など

SFFファイルの構造

拙著 奮闘記でも書いたことがあります ([Win版における色化け対策というか](#)) が、このソフトを開発するにあたり気づいた点・執筆当時誤解していた点などがありましたので、改めて記します。

SFF フォーマット概念図

[SFF ヘッダ]	[サブファイル]		[サブファイル]
	[サブ ヘッダ]	[PCX データ]	[サブ ヘッダ]
登録イメージ数 イメージデータ開始アドレス パレットモード区分	GroupNo ImageNo axis X axis Y パレット区分 次サブファイル開始アドレス		GroupNo ImageNo axis X axis Y パレット区分 次サブファイ...
		[パレット]	

[SFF ヘッダ]

SFF ファイルの先頭512バイトがヘッダ部となっています。

SFF 全体に関わる情報が格納されています。

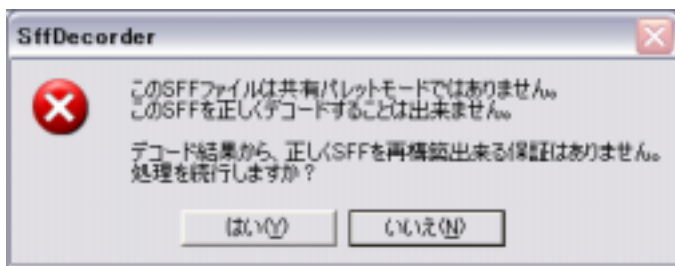
ここで曲者なのが「パレットモード区分」です。

これは、SprMaker でSFFを作成する際に、最初に指定できるオプションです。

通常、「共有パレットモード」を指定して作成されますが、希に「独立パレットモード」にて作成されているSFFが存在します。

SFFDecoder では、「共有パレットモード」しかサポートしていません。

「独立パレットモード」で作成されたSFFを読み込ませようとすると、次のメッセージが表示されます。



一応、解析結果は表示されますし、SFF再構築用のデータも出力できますが、バッチファイルを実行してもSFFは正しく再作成されません。

これは、SFF作成用のパラメータテキストが、独立パレットモード用ではないためです。

ダークチュンリーのSFFが独立パレットモードで作成されていたので、確認してみたところ、

再作成されたSFFをmcm2.xで確認すると、一部の共有パレットスプライトが独立パレットとして登録されてしまっていました。

なお、パレットモードに関わらず、再作成したSFFをmcm1.3で読み込ませると、アプリケーションエラーが発生することがあります。mcm2.xであれば大丈夫ですので、そちらを使用して下さい。

[サブファイル]

ヘッダ部に続いて、各スプライトデータが続きます。

それぞれにサブヘッダ(32バイト)が付加され、個々の情報が記録されています。

サブヘッダの後にPCXデータがそのまま記録されています。

バイナリエディタなどでPCXデータに該当する部位のみを切り出して保存すれば、PCXファイルとして扱えます。

(ただし、共有パレットで登録された画像からは、パレット情報が欠落しています。)

登録された画像数分、サブファイルが存在します。

「共有パレット」の幻想

mcmに慣れている人にとっては、最も誤解を招いているであろう言葉がこの「共有パレット」です。

mcmで画像を追加する際に、「Use Shared Palette」「同じパレットを使う」などのオプションを指定した場合、この共有パレットとなります。

ただ、SFF自体には、全体で共通的に使用するパレット情報が特別に保存されているわけではありません。

それは上記の構造図をご覧いただければお分かりでしょう。

では、「共有パレット」とは何なのかと申しますと、

実は、「直前に登録された画像と同じパレットを使用する」という意味なのです。

つまり、こういうことです。

[SFF ヘッダ]	[サブファイル]	[サブファイル]	[サブファイル]	[サブファイル]
	パレット情報有	パレット情報なし のパレットを使用	パレット情報なし のパレットを使用	パレット情報なし のパレットを使用

上図は、SFFに保存されたデータの概念図です。

全ての画像は「共有パレット」として登録されているものとします。

「共有パレット」として登録された画像は、登録時にパレット情報そのものが省略されます。

そして、一つ前に登録されている画像のパレットを参照するよう、フラグがセットされています。

(但し、共有パレット画像であっても最初に登録した画像に限り、パレット情報を保持しています。)

ここで、例えばMUGENがサブファイル の画像を参照しようとした場合、そのパレットは「直前に登録された画像と同じ」であるため、

一つ前の画像、つまりサブファイル のパレットを見に行きます。

ところがサブファイル も「直前に登録された画像と同じ」パレットを使用するため、さらにその一つ前の画像、つまりサブファイル のパレットを参照することになります。

Win版MUGENにおいて、SFFの途中に独自パレットが存在する場合に色化けが発生する原因は、この仕様にあります。

[SFF ヘッダ]	[サブファイル]	[サブファイル]	[サブファイル]	[サブファイル]
	パレット情報有	パレット情報なし のパレットを使用	パレット情報有	パレット情報なし のパレットを使用

上図は、SFFの途中に独自パレットが存在する場合の概念図です。

サブファイル が独自パレットであるため、サブファイル はサブファイル のパレットを参照してしまいます。

本来の色情報で画像を表示するためには、サブファイル のパレットを参照する必要がありますから、結果としてサブファイル の画像は色化けをおこして表示されてしまうことになります。

サブファイル や が共有パレットである場合も、結局 のパレットを参照してしまうため、以後に登録された画像が色化けをおこします。共有パレットを使用する画像は、ひとまとめに連続して登録しておかなければならないのは、この理由によります。

さて、このSffDecorder の存在理由ですが、

色化けを避けるために、SFF前半に共有パレット画像をひとまとめに連続して登録し、その後、大量の独自パレット画像を登録してある場合に、共有パレット画像を追加したい場合、mcmではどうやりますか？

mcmでは登録画像の物理的保存順位は変更できませんから、独自パレット画像を一度全て削除して、共有パレット画像を追加、その後、独自パレット画像を全て再登録、という作業になるでしょう。

独自パレット画像が何十枚も登録されていたなら、ちょっと勘弁して欲しい作業となりますよね。

そこで、SFFDecorder の出番です。

とりあえずmcmでx、y基準点を調整しながら共有パレット画像を追加しましょう。

mcmでは、追加画像は(物理的には)ファイルの最後に追加されますから、そのままではWin版MUGENで色化けをおこしてしまいます。

なので、SFFDecorderでSFF再構成に必要なデータを出力し、生成されたパッチファイルを実行してSFFを作り直してしまいましょう。

SprMaker用パラメータテキストを出力する際に、画像の登録順序を自動的に変更してくれる(共有パレット分を前半に固めてくれる)のでWin版MUGENでの色化けを避けることができます。

ポートレート画像のワナ

グループ番号 9000 番の、いわゆるポートレート画像(小・大)については、実はもう少し注意が必要です。

これまで述べてきた通り、共有パレットで登録された画像データからは(1枚目を除き)、物理的にはパレット情報が削除されています。

通常の画像であれば問題ないのですが、ポートレートに関しては要注意です。

特に、共有パレットとしてポートレート画像を登録する場合は気をつけて下さい。

WinMugen のソースコードを覗いた訳ではないので間違いがあるかもしれませんが、どうやらポートレート画像には(パレットモードに関わらず)パレットデータが**物理的に内包されている必要がある**ようです。

独自パレットとして登録する分には、物理的にサブファイル内の画像データにパレット情報データが含まれるので問題ないですが、共有パレットの場合、最初に共有パレットとして登録された画像にのみ、パレット情報データが格納され、以外は先に登録された画像への参照アドレスがセットされるのみで、物理的にはパレット情報データが存在しない状態となります。

[SFF ヘッダ]	[サブファイル]	[サブファイル]	[サブファイル]	[サブファイル]
	Group:0	G:0	G:9000	G:9000
	Item:0	I:1	I:0	I:1
	パレット情報有	パレット情報なし のパレットを使用	パレット情報なし のパレットを使用	パレット情報なし のパレットを使用

この状態でWinMugenにてポートレートを表示すると、パレットデータが存在しない画像として扱われ、結果、色化けを起こすことになってしまいます。

共有パレットとしてポートレート画像を登録するのであれば、SFFの先頭に登録してやると良いでしょう。

先頭に登録された共有パレット画像には、物理的にパレット情報データが格納されますので。

[SFF ヘッダ]	[サブファイル]	[サブファイル]	[サブファイル]	[サブファイル]
	Group:9000	G:9000	G:0	G:0
	Item:0	I:1	I:0	I:1
	パレット情報有	パレット情報なし のパレットを使用	パレット情報なし のパレットを使用	パレット情報なし のパレットを使用

上図の状態であれば、私の環境で確認した限り、ポートレートが色化けすることはありませんでした。

或いは、ポートレート画像自体を独自パレットとして登録しておくのも手です。

以上を踏まえ、SffDecorder では、処理に汎用性を持たせるため、下図の様な構成にてSFFを再構築できるよう、パラメータを設定します。

[SFF ヘッダ]	[サブファイル]	独自	[サブファイル]	独自	[サブファイル]	独自	[サブファイル]	共有
	Group:9000	パレット情報有	Group:9000	パレット情報有	Group:0	パレット情報有	G:0	パレット情報なし
	Item:0		Item:1		Item:0		I:1	のパレットを使用

グループ0のアイテム0が独自パレットとして登録されている事に違和感を覚えられるかもしれませんが、「共有パレット」という言葉に惑わされないで下さい。

ElecbyteによるSFFファイル仕様に従えば、登録する画像感のパレットが統一されていれば、この(物理的)データ格納方法で問題ありません。

余談ですが、mcmで画像のコピー & 削除を繰り返すとSFFが潰れた、といった話を良く耳にしますが、もしかしたら、「物理的にパレット情報を保持している画像」を誤って消してしまうことにより、共有パレット情報が破壊されてしまった為に潰れるのかもしれないね。

mcmの内部処理がどうなっているのか、ちょっと分からないのでハッキリしたことは言えませんが、^-^-;

- 以上 -

ご参考

キャラ制作奮闘記

[「Win版時代のSFF構築術 ~ sprmaker活用法 ~」](#)

[「Win版での色化け対策というか」](#)

[「SNDファイルを作ってみよう」](#)



(C) raiwa 2005 All Rights Reserved