

アニメプロップ製作指南書

By denham2010 – December 20, 2005, Simtropolis STEX #14283

アニメプロップは複雑なプロセスで作製されます。それは決して困難な作業ではないものの、**100枚以上のビットマップ処理、長時間の忍耐、集中力は確実に必要**となります。

ここでは手順を逐一記載するよりも、回転するサインの作製を通して解説していきます。このダウンロードファイルには、読者がオリジナルを作製する際にテンプレートとして活用できるよう、**animconstructpack** というファイルも同梱しています。

この指南書では製作手順を以下の章に分けて解説しています。

1. アニメーションシークエンスの製作
2. フレームビットマップの抽出
3. アニメーションビットマップの作成
4. SC4ファイルの改造（.datの作製）

手順1と4は極めて単純で、簡単に処理できます。

手順2と3は困難ではありませんが、集中力や心構えが充分でないと、いらだちを感じたり、時間を浪費しているように思えるでしょう。

作製を開始するに当たっては以下のツールが必要となります。

1. Gmax/BAT
2. FiSHMan
3. llive Reader
4. 画像処理ソフト（Photoshop または同等品）
5. Lot Editor

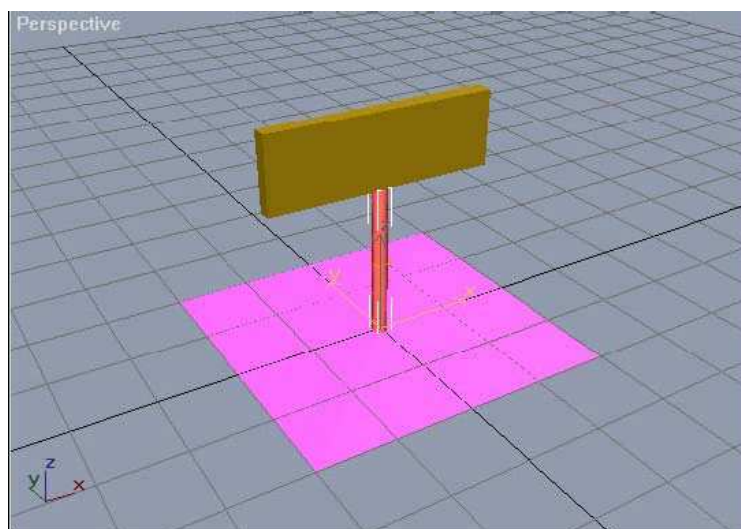
これらは画像処理ソフトを除いて、**Simtropolis** または **SC4公式サイト**でダウンロードできます。

では、始めましょう。

1a. アニメーションシークエンスの製作

この指南書では、読者にBAT製作の知識が既にあることを前提にしています。もし、まだ知識が充分でなければ、多くの優れた指南書がSimtropolisにありますので、始めるに当たって参考にしてください。

最初に、次のモデルを作製します。柱の上にサインが乗っている簡単なモデルです。



以下の要領でモデルを作ってください。

1. 半径0.1m、高さ2.0mの円筒で柱を作り、柱の中心をX,Y軸の原点に合わせます。
2. 幅3.0m、高さ1.0m、奥行き0.2mの箱を作り、箱の底の中心をX軸の原点に合わせます。Y軸は0.1m（奥行きが負の場合は-0.1m）の位置に中心を合わせます。そして、箱をZ軸方向に2.5m持ち上げます。
3. 4.0m×4.0mの四角形をつくり、中心をX,Y軸の原点に合わせます（この四角形の色は強調されていますが、これは必要なことではなく、また、この四角形はモデルの一部でもありません）。
4. 箱と柱にUVWマップを当て、無地の白のみでテクスチャを貼ります。また、それらの **self illuminating** にも無地の白を当ててください。

4.0m角の四角形が何であるのか不思議に思われるでしょうが、この四角形の唯一の目的は、アニメーションフレーム（コマ）を互いに確実に並べられるようにすることです。この四角形は少なくとも全てのアニメーションの幅と深さを満たしている必要がありますので、作成されるすべてのフレームを考慮に入れなければなりません。この四角形は正方形にするのが原則です。正方形とすることで、各ステップがずっと容易になります。また、同時にアニメーションの高さも考慮する必要があります。なるべく高さは幅と奥行きの1/4以下となるようにしてください。これらは不要なことですが、作成するビットマップが正方形の範囲を保つことで、アニメの製作が簡便になるからです。【コメント01】

この四角形にも UVW map を当て、transparent は100%（透明）にします。

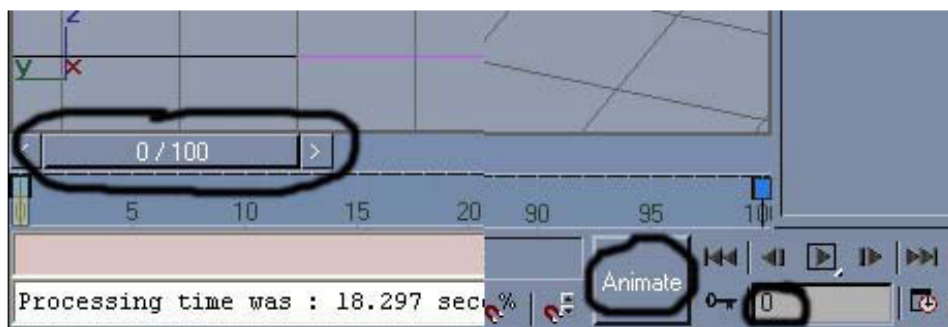
1b. モデルをアニメートする

これから、作成したサインに動きを与えます。しかし、その前にサインの**Pivot Point**（回転基点）を変える必要があります（スクリーン上のX,Y,Z軸の交点を移動しようとするとき選択するのが**Pivot Point** --- 周囲の動き全ての基準となる点です）。



1. 箱を選択し**HIERARCHY**タブをクリックします。
2. そこの、**AFFECT PIVOT ONLY**ボタンを押します。
3. ツールバーの**SELECT AND MOVE**ボタンを押します。そして、**Pivot Point**のY座標を**0.0 m**に合わせます（正確に合わせるため、ステータスバーの**XYZ** テキストボックスを使ってください）。
4. そうしたら、**MODIFY**タブ（**HIERARCHY**タブの左側）をクリックして、**Pivot**モードを抜けます。

これで、回転軸は正確にセットされたので、アニメーションが可能となります。



箱が選択されていることを確認してください（**PERSPECTIVE**窓の中で）。そして、ツールバーの**SELECT AND ROTATE**ボタンを押します。

1. ステータスバーの**ANIMATE**ボタンを押します。上のタイムスライダが赤になります。
2. それから、スライダをフレーム数**25**の位置にします（または、**ANIMATE**ボタンの隣のテキストボックスに**25**と入力してください）。

3. 箱をZ軸周りに90度の位置に回転します。
4. スライダを50にし、箱を180度の位置にします。
5. スライダを75にし、箱を270度の位置にします。
6. 最後にスライダを100にし、箱を360度の位置にします。
7. もう一度ANIMATEボタンを押してフレームの記録を停止します。

注意しますが、セットした最後の回転角は360度であって0度ではありません。Gmaxにはひどい癖があって、回転角を増加していった後に小さな回転角をセットすると、アニメーションの向きを逆転させてしまいます。この現象は起きたり起きなかったりですが、安全策をとるべきでしょう。

作成したモデルを **Sign.gmax** (既存の同名は無いはず) という名前で保存してください。

お楽しみとして、**ANIMATE**ボタンの右側の、**PLAY** (四角で囲まれた三角形) ボタンを押し、そのサインが回転するのを見てください。同じボタンを押すと、アニメーションが止まります。

タイムスライダを0に戻します。

次に、モデルのLODSをセットアップする必要があります。ここで、上で作った四角形の出番となります。

1. メニューバーの**FILE** (ファイル) をクリックし、**XREF SCENE** (外部参照シーン) を選択します。
2. 枠の中の **lighting rig** 参照先を選択し、**REMOVE**ボタンで削除します。そして、ダイアログを閉じます。
3. **UTILITIES**タブ (モデルをエクスポートするときと同じタブ) から**BAT**ボタンを押し、**PARAMETERS**ロールアウトを選択します。
4. **REFIT LODS**ボタンと、**RESET LIGHTS AND CAMERAS**ボタンを押します。

ここで、モデルのLODSが現れるようにしましょう。このLODSの中にアニメーションの全てが納まるようにしなくてはなりません。上の四角形はXとY軸について範囲を確実にします。また今回は、アニメーションの高さが変化しないので、サインの頂部は問題がありませんが、そうでない場合は、各々のLODの高さをアニメーションが使う最大の高さに調整してください。

重要なことですが、アニメーションの範囲が収まるようにLODSをセットし、各々のフレーム (コマ) でLODSを変化させないようにしてください。さもないと、アニメーションpropは、ロット上のセットされた位置に留まらず、ロットの外に飛び出してしまうでしょう。

それでは、次のお楽しみを初めましょう。

1. モデルをエクスポートし、ファイル名を **1A(.gmax)** に変えます。
2. タイムスライダを10の位置にセットし、再びエクスポートし、ファイル名を **2A** に変えます。

3. 10フレーム（10コマ）エクスポートするまで、上の操作を繰り返します（スライダは90を指しています）。
4. スライダを0に戻し、モデルに（サインに見えるような）テクスチャを貼ります。そして、上と同じエクスポート手順を繰り返しますが、今回はモデルの名前を1,2,3などにします。

ここまでで、10個のフレームモデルと10個のアルファモデルが完成しているはずです。

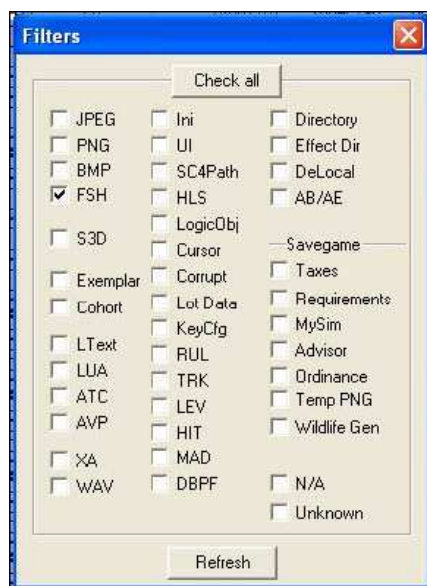
今回の例では、できるだけ単純にするため10フレーム（コマ）のみを使っています。もっと多くのフレームを使えば、アニメーションはより滑らかになるでしょう。20－24フレームが理想的な値です。

2. フレームビットマップの抽出

全体の手順の中で、ここでの作業は時間のかかる退屈なものとなります。気持ちを集中させて、どのフレーム（コマ）も混ぜてしまわないよう注意してください。

ILive Reader を起動します。そして、最初のフレームモデル（1.SC4Model）を開きます。

1. ツールバーの**FILTERS**ボタンを押してください。



2. ダイアログの**UNCHECK ALL**ボタンを押します（画面では**CHECK ALL**）。
3. **FSH**にチェックを入れて、**REFRESH**ボタンを押し、ダイアログを閉じます。
4. モデルの**.FSH**ファイルのみが表示されているはずです。
5. ヘッダーの**INSTANCE**をクリックし、**.FSH**ファイルが番号の低い方から順に表示されるようにします。

Entry	Com...	Filesize	Loca...	Num	Type	Group	Instance
FSH file	Y	75	8660	7	7ab50e44	1C6ec7a8	00220000
FSH file	Y	80	8995	16	7ab50e44	1C6ec7a8	00220010
FSH file	Y	75	9344	20	7ab50e44	1C6ec7a8	00220020
FSH file	Y	80	9698	24	7ab50e44	1C6ec7a8	00220030
FSH file	Y	85	7233	28	7ab50e44	1C6ec7a8	00220100
FSH file	Y	80	7586	32	7ab50e44	1C6ec7a8	00220110
FSH file	Y	85	7938	36	7ab50e44	1C6ec7a8	00220120
FSH file	Y	80	8306	41	7ab50e44	1C6ec7a8	00220130
FSH file	Y	112	5667	43	7ab50e44	1C6ec7a8	00220200
FSH file	Y	104	6058	45	7ab50e44	1C6ec7a8	00220210
FSH file	Y	112	6446	1	7ab50e44	1C6ec7a8	00220220
FSH file	Y	104	6847	3	7ab50e44	1C6ec7a8	00220230
FSH file	Y	163	3899	5	7ab50e44	1C6ec7a8	00220300
FSH file	Y	143	4347	14	7ab50e44	1C6ec7a8	00220310
FSH file	Y	163	4782	18	7ab50e44	1C6ec7a8	00220320
FSH file	Y	143	5236	22	7ab50e44	1C6ec7a8	00220330
FSH file	Y	323	1509	26	7ab50e44	1C6ec7a8	00220400
FSH file	Y	300	2113	30	7ab50e44	1C6ec7a8	00220410
FSH file	Y	323	2700	34	7ab50e44	1C6ec7a8	00220420
FSH file	Y	300	3315	38	7ab50e44	1C6ec7a8	00220430

ここで、.FSHファイルの説明をしておきます。図には20個のファイルが表示されています。SC4には5つのズームと4つの回転視点があります。それぞれのファイルは1つのズームと1つの回転視点を表現しています。

Instance IDの最後の3桁は、以下のようにズームレベルと回転視点位置を参照しています：

0XX ズーム1（最遠景）
 1XX ズーム2
 2XX ズーム3
 3XX ズーム4
 4XX ズーム5（最近景）
 X00 南面画像
 X10 東面画像
 X20 北面画像
 X30 西面画像

ここで重要なのは、確実な命名規則の使用で、順序良くファイルを抽出できるようにすることです。次の章で特定の順番にそれらを再構築するときに意味を持ちます。

さて、次はアニメプロップを、SC4で、どのズームレベルまで見えるようにするかを決めます。レベルの決定はプロップのサイズに依存します。小さなプロップではズームレベルは3つで足りませんが、大きなものは5つ全部が必要となります。今回の例では、サインは4つのズームレベルで足りません。しかし、ここでは、3つに留めることにします。

分かると思いますが、3ズームレベルと4回転視点と10フレームでは計120個のビットマップとなり、アルファビットマップも同様に抽出するため、240個のビットマップを扱うことになります（なぜ、ここでの作業が時間を必要とし、効率の良い命名規則が大事なかが分かるでしょう）。

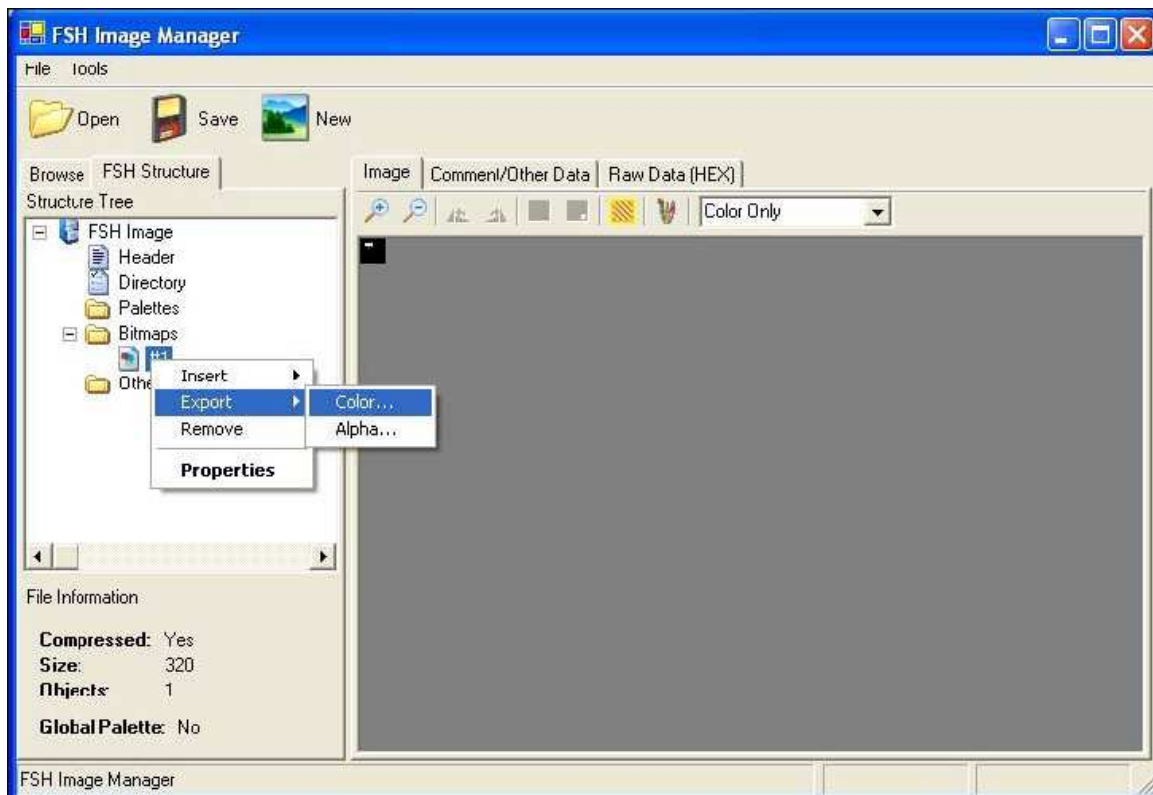
作業用にファイル構造の作成を推奨します。ズーム毎のフォルダに、各回転視点の下位フォルダを付け、さらにモデルとアルファチャンネル用の下位フォルダをつなげます。

1. 最初のズームレベルの南面画像用.FSHファイル（上の例では、Instance IDがXXXXXX200 --- ズーム3、南面画像）をクリックします。
2. シフトキーを押したまま、下矢印で表の下まで残りのファイルを選択します（重要 … 選択した順番にファイルが保存されるので、必ず表は番号の低いほうから表示されるようにしてください）。
3. 選択された部分の上で右クリックし、**SAVE SELECTED FILE(S)**をクリックします（注：自動保存を止めるため、トップメニューのoption / optionsで"ask for filename when saving"にチェックを入れておいてください）。
4. ズーム3南面モデルのフォルダを指定し、ファイル名を **1S3.FSH**として保存します（必ず拡張子.FSHをファイル名の最後につけてください）。
5. iLive Readerは自動的に次の選択ファイルの保存を準備します。今度はズーム3東面モデルとなり、ファイル名**1E3.FSH**で保存します。
6. 次は北面（**1N3.FSH**）、そして西面（**1W3.FSH**）を保存します。
7. 続いて、次のズームレベル（この例では4つ）を**1S4, 1E4, 1N4, 1W4**として保存します（.FSHを付け忘れないように）。
8. ズーム5についても同様に保存します。
9. フレームモデル（**1.SC4Model**）を閉じ、次のフレームモデル（**2.SC4Model**）を開きます。
10. 上のステップを繰り返します（今回は、**2S3, 2E3....2N5, 2W5.FSH** として保存します）。
11. そうして、すべてのフレームモデルが保存されるまで処理を続けます。
12. さらに、アルファモデルについても同様に処理を繰り返します。ここではファイル名に**A**を付加します（**1SA3.FSH**）。

ビットマップの抽出は、道のりの半分までできました。

FiSHManを起動します。ここで必要とされる作業は、各々のフォルダから実際のビットマップを引き出すことです。上のステップと同様に同じ順を追って行きます。

1. ツールバーの**OPEN**をクリックし、一番低い（遠い）ズームから南面モデルのフォルダを指定し、最初のフレームを開きます（この例では**1S3.FSH**）。



2. FSH Structureタブを選び、Structure Treeのビットマップ#1をクリックします。
3. もう一度その#1を今度は右クリックし、EXPORT / COLORを選択します
(Exportオプションは最初に1回左クリックするまで表示されません)。
4. ビットマップをFSHファイルと同じフォルダに同名で保存します (SAVE AS
TYPEオプションをWINDOWS BITMAPに変えるのを忘れないように)。
5. 抽出した全てのFSHファイルに対して、この操作を繰り返します (注：アルファ
ファイルのときは、ステップ3で EXPORT / ALPHA を選択します)。

最後まで終わりましたか？少し時間が必要ですね。さて、これまでに（この例では）24
セットのアニメーションフレームが揃い、それらを繋ぎ合わせる用意ができていますはずです。
ズーム3南面モデルフォルダには、1S3.BMP, 2S3.BMP...9S3.BMP, 10S3.BMPが存在し、そ
のアルファフォルダには1SA3.BMP.....10SA3.BMPが存在しているはずです。これは、残り
の東面、北面、西面のフォルダと他のズームレベルについても同じです。後で困ったことにな
らないよう、どのファイルも失くさないようにしてください。

さて、ここで一息つきましょう。想定外の恐ろしい仕事が待っていますよ。次はこのジ
グソーピースを繋ぎ合わせることになります。

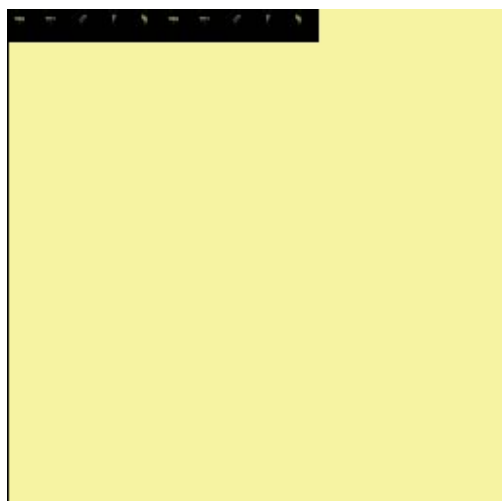
【コメント02】

3. アニメーションビットマップの作成

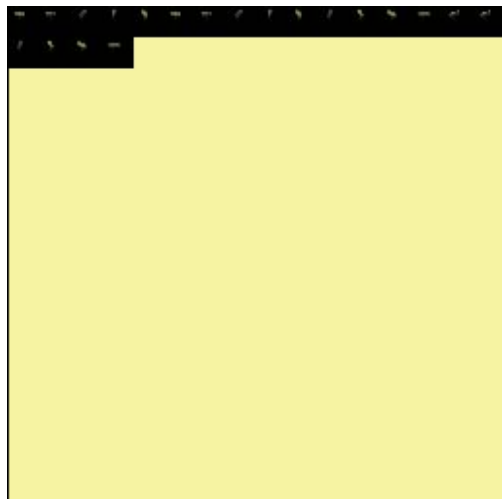
このたくさんのピースをどう繋ぎ合わせたらいいのか？ それには、まず、**SC4**がファイルを一方向にしか読み込まないことに注意し、確実にファイルを正しい順序で配置することです。そうしないと、例えば南と東の回転像が間違った位置に表示されたり、フレームの順番を間違えれば、アニメーションはコマ飛びしたり、ロットに正しく座らなくなる原因となります。

ここで、フォトエディタを起動してください（ここでは**photoshop**を使っています。推奨はしませんが、ペイントでもかまいません）。この作業の間、ノートパッドも同様に起動しておきます。いくつか計算を書き留めておく必要が出てきます。

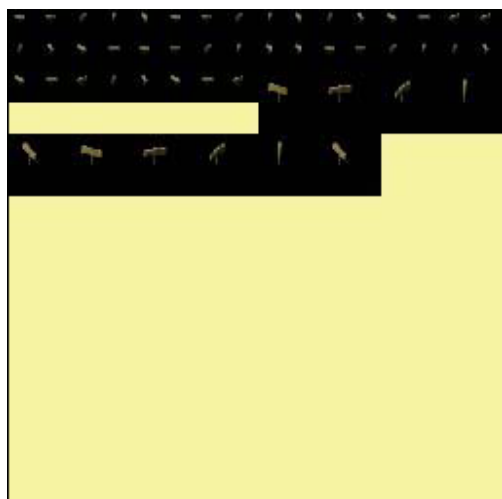
1. フォトエディタの中に、最も低いズームの南面モデルフォルダ（例ではズーム**3**）から、全てのファイルを開いてください（**1S3** から **10S3** までの**10**個）。
2. **256×256**ピクセルのビットマップを新規に作ります。そして、バックグラウンドを黒や白以外の色にセットします。
3. それぞれのフレーム（コマ）を、新しいビットマップにコピー&ペーストで写していきます。互いに正しく隣り合わせます（絶対オーバーラップしないよう注意してください。ここでは厳格な人になりましょう）。
4. ノートパッドには **0 0 16 16 8 8** と記録します（これらの数値については次章で説明します）。



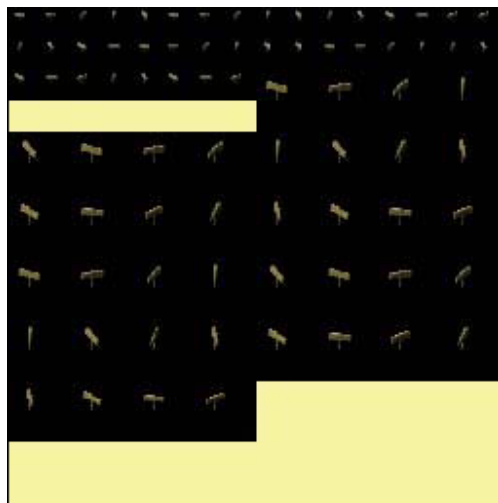
5. 次に、最も低いズームの東面モデルを開きます（例では **1E3** から **10E3**）。
- そして、上のビットマップに、南面ファイルの終わりに続けるように貼り付けます（行の終端に達したら次の行に移ります。もし、写す画像が終端に合わない場合は、空白を残して次の行に移ってください）。
6. ノートパッドに **0 160 16 16 8 8** と記録します。



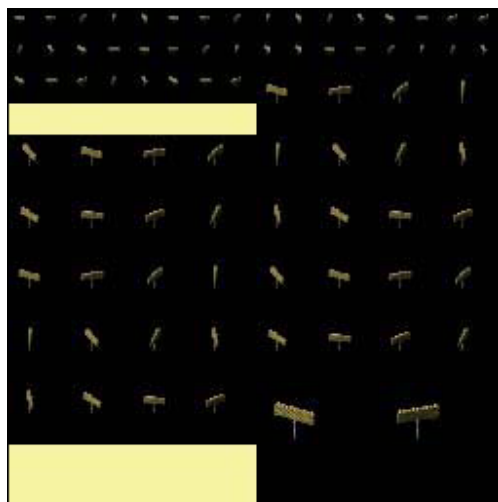
7. 続いて北面の画像を貼ります（ 1N3 から 10N3 ）。そして、0 4160 16 16 8 8 をノートパッドに記録します。
8. さらに続けて西面の画像を貼り（ 1W3 から 10W3 ）、0 4320 16 16 8 8 を記録します。
9. 次に、次のズームの南面ファイルを開きます（ 例では、1S4 から 10S4 ）。そして、オーバーラップ無しに画像の貼り付けを続けます（今回は新しい行を始めるとき、上にギャップを入れる必要が出てきます。そのギャップによって、その行の貼り付けが進んでもオーバーラップは発生しません --- この例では、最初のズームのファイルが全て16×16ピクセルなので、ギャップは16ピクセルとなります）。
10. ノートパッドには 0 8320 32 32 16 16 と記録します。



11. 続けて東面のファイル进行处理し、さらに北面、西面と続けます。各面の記録は 0 16576 32 32 16 16 と 0 32768 32 32 16 16 と 0 41024 32 32 16 16 になります。
12. 次に何をするかというと、最後のズームレベル进行处理する仕事がまだ残っています。しかし、もうビットマップに余地がありません。でも心配御無用、ノートパッドに 0 49280 64 64 32 32 と記録してから、最終ズームの南面ファイル（1S5 から 10S5）を開き、最初の2枚を貼り付けることができます。

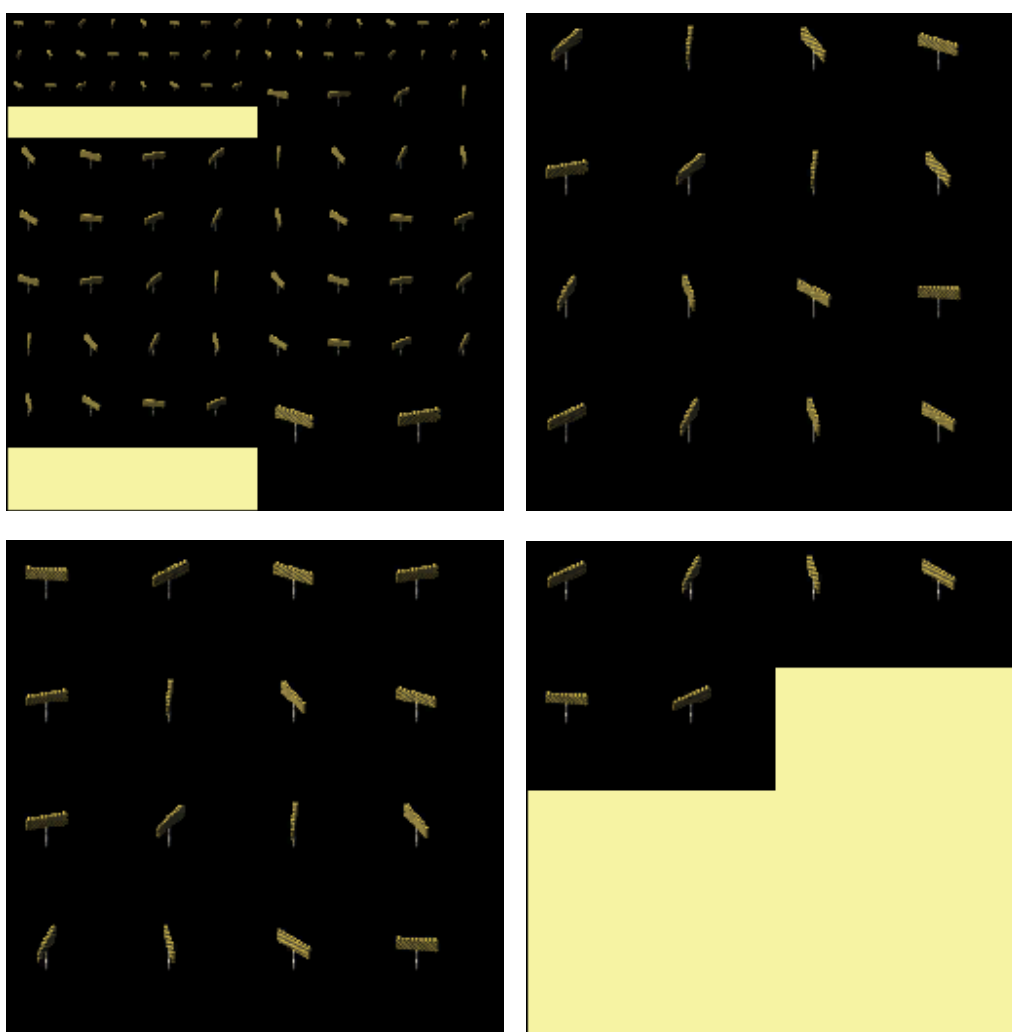


13. 最初の2枚を貼り終えたら、**#1**の名前でビットマップを保存します。名前はただ**#1**だけでかまいませんが、必ず 8 bits/channel で Windows Bitmap または PNG として保存してください。
14. もう1枚の新しいビットマップを**256×256**ピクセルで作ります。そして、残りの画像の貼り付けを続けます。



15. 東面ファイルの処理を続け、**1 32768 64 64 32 32** と記録します。8枚貼り付けると、再びビットマップに余地が無くなるので、そのビットマップを **8 bits/channel** の PNG または **Bitmap** として保存します。名前は **#2** とします。
16. 3番目の**256×256** ブランクビットマップを作り、残った東面画像2枚を貼ります。続けて北面を処理します。記録は **2 128 64 64 32 32** です。
17. このビットマップには西面を4枚貼り付けられます。他の6枚は新しいビットマップに貼り付けます。最初の4枚を貼り付けたら、**2 49152 64 64 32 32** と記録します。ビットマップは **#3** の名前で保存します。残った6枚も新しいビットマップに貼り、**#4** として保存します。

こうして、以下のように4枚のビットマップが揃います：



このプロセスをアルファファイルについても行い、4枚のビットマップを **#1A, #2A, #3A, #4A** として保存します。今回は数値をノートパッドに記録する必要はありません。1セットの数値で足ります。ノートパッドの記録は **filepos.txt** の名前で保存します。【コメント03】

5. SC4ファイルの改造

アニメーションプロップ製作の最終工程に入ります。作成したSC4ファイルを一つの.datファイルにまとめ、プラグインフォルダに放り込んでロット製作で使えるようにします。

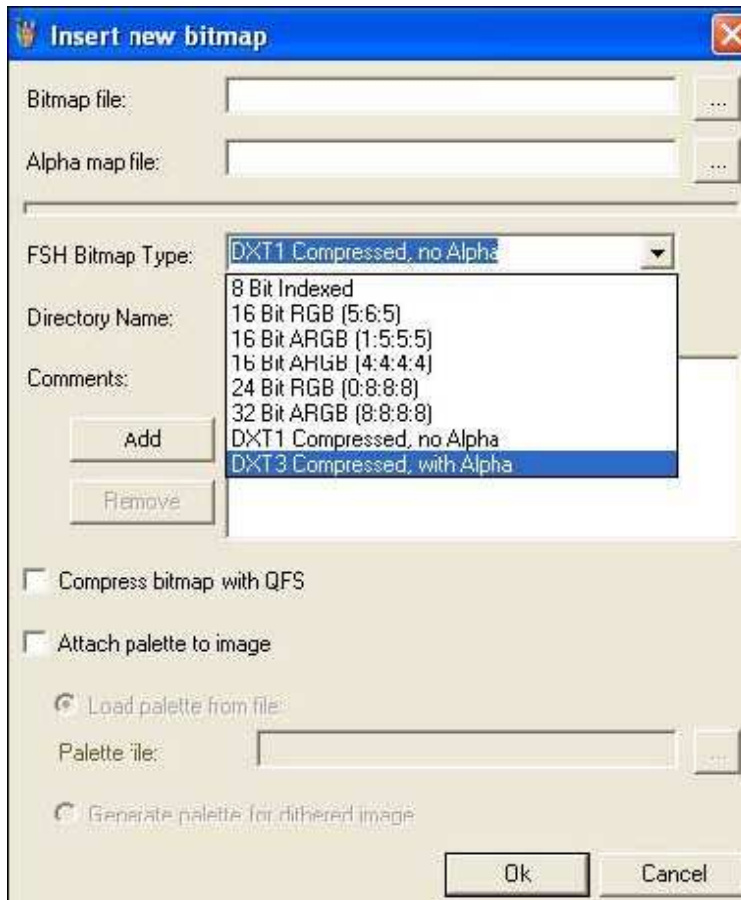
この指南書に同梱されている建設パックには、次のファイルが含まれています。

1. animconstructpack.dat
2. animconstructpack.fsh.TGI

Ilive Readerで上の.datファイルを開くと分りますが、このファイルは5つのAVPファイルと、それぞれ1つの、ATCファイル、FSHファイル、Directoryファイル、Exemplarファイルから構成されています。

最初のステップは.FSHファイルを今回作成したものと置き換える作業です。

1. animconstructpack.fsh.TGIをコピーし、名前をsign.fsh.TGIに変え、ノートパッドで開きます。
2. これはアニメFSHファイル用の、typeとgroupとInstanceのファイルです。
最初の値(7AB50E44)はtype値で、このファイルが.FSHとしてSC4に参照されます。二番目の値は(49A593E7)はgroup値で、このFSHがprop FSHであることSC4に伝えます。オプション値は2つあり(49A593E7 と 2A2458F9)、どちらも交換可能ですが、.datファイルがどこから参照されても同じ値を保つ必要があります。
最後の値(FFFFFF00)は、このアニメプロップのInstance IDです。値はデフォルト値なので、覚えやすい値に変える必要があるでしょう。SC4は一般に上位2バイトを使用します。3番目のバイトは通常00で、最下位バイトは必ず00です。SC4が使用しているinstance IDの最大値は2C360000なので、それ以上の値を使用するのが最適です。
3. ここではEEFF0000を使用します。ノートパッドのInstance ID値をEEFF0000に変え、保存します。
4. FiSHManを起動します。ファイルメニューでNEWを選択し、FSH STRUCTUREタブをクリックして、ポップアップメニューのBITMAPを右クリックします。
5. IMPORT BITMAPを選択し、FSH Bitmap TypeをDXT3 Compressed with Alphaに変更します。
6. Bitmap file欄では、最初のPLANE(面:#1.bmpで保存)を参照選択します。
7. Alpha map file欄では、最初のALPHA PLANE(#1A.bmpで保存)を選択します。
8. OKボタンを押します。
9. ステップ5から8までを、4枚のビットマップとアルファチャンネルの全てが順番にインポートされるまで繰り返します。



10. ツリーに4枚のビットマップが表示されたら、ファイルを**sign.fsh**の名前で保存します（固有名**sign**は**fsh.TGI**と同じにしてください。**FiSHMan**がこの**.FSH**ファイルを参照すると、**TGI**ファイルにも同名を割当てます）。

これで反復アニメーション用の**.FSH**ファイルができました。

ここで作業を続ける前に、数値コンバートが少々必要となります。前章でノートパッドに記録した数値を思い出してください（**0 0 16 16 8 8**等）。この数値は**FILEPOS.TXT**として保存されてますよね？この数値を**16進値**に変換すると、**AVP**ファイルに記入できるようになります。

AVPファイルは**SC4**が、**.FSH**ファイルの中で各々のズームと回転視点について、アニメーションの開始点を参照するために使用します。

記録された最初の数値は**PLANE**です。**.FSH**ファイルにアップロードされた**#1, #2, #3, #4**として参照されるビットマップ面です。**SC4**はそれらを**#0, #1, #2, #3**として参照します。この値でシーケンス（1ズーム、1回転視点でのアニメーション。ここの例では**10**フレームのビットマップセット）が正しく開始するためのビットマップを指定します。

二番目の値はFILEPOSです。この値は最初のシークエンスの開始点までに、ファイルの中に何ビットあるか（ビット列の中でのシークエンスの開始点）を参照します。当然、最初のシークエンスの値は0です。ここの例では、二番目のシークエンスの値は、1シークエンスが10フレームで、1フレームが16ビット幅なので、160となっています。三番目のシークエンスでの値は4160となっていますが、このフレーム行は2行目なので、上に16ビット行あり、各ビット行は256ピクセル幅なので（ $16 \times 256 = 4096$ ）となり、そして、すでに4フレームが前にあるので（ $4 \times 16 = 64$ ）なので $64 + 4096 = 4160$ ）となります。

計算するときは、高いズームに行くほどフレームサイズが大きくなることを憶えておきましょう（ズーム4で 32×32 、ズーム5で 64×64 ）。

三番目と四番目の値はシークエンスのフレームサイズで、幅と高さの順になります。幅と高さは同じ値である必要はありませんが、様々な‘prop float’の被害を避けるためには、できるだけ正方形を保つべきです。

最後の2つの値はフレームのオフセットです。SC4がプロップを配置するときに使用し、ロットエディタで表示されるプロップ設置マークの位置です。プロップの中心はフレームの左上の角となります。そのため、もしオフセットを0のままにすると、視点を変えるたびにプロップはロットの周りを移動してしまいます。プロップを正しく揃える最も良い方法は、オフセットをフレームの高さと幅の半分にすることです。もしプロップが正方形でない場合には、少々調整したほうが良いでしょう。

とにかく、フレームを正方形に保てば人生はとても楽になり、‘prop float’の被害も大幅に少なくなります。

注：1つのシークエンスの全てのフレームは同じサイズでなくてはなりません … 各々のズームと回転視点について使用するフレームサイズは1つのみです。

10進数を16進数に変換しましょう。結果はこのようになります：

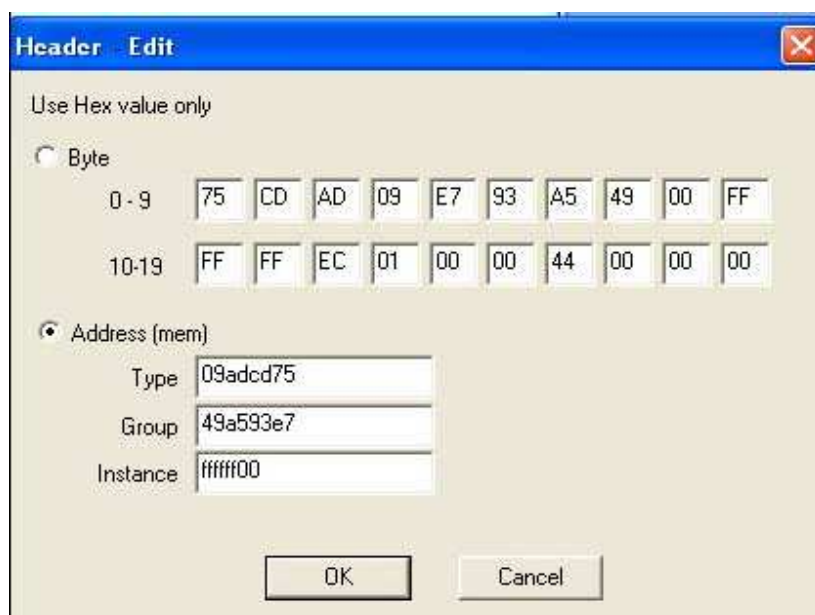
0	0	16	16	8	8	00	0000	10	10	08	08
0	160	16	16	8	8	00	00A0	10	10	08	08
0	4160	16	16	8	8	00	1040	10	10	08	08
0	4320	16	16	8	8	00	10E0	10	10	08	08
0	8320	32	32	16	16	00	2080	20	20	10	10
0	16576	32	32	16	16	00	40C0	20	20	10	10
0	32768	32	32	16	16	00	8000	20	20	10	10
0	41024	32	32	16	16	00	A040	20	20	10	10
0	49280	64	64	32	32	00	C080	40	40	20	20
1	32768	64	64	32	32	01	8000	40	40	20	20
2	128	64	64	32	32	02	0080	40	40	20	20
2	49152	64	64	32	32	02	C000	40	40	20	20

llive Readerを起動し、animconstructpack.datのコピー（sign.dat）を開きます。

最後のバイトが00から04となっている5つのAVPファイルが見えます。これらはゲーム中のズームレベルを参照しています。この例では3つのズームのみを使用しているので、必要とするAVPファイルは 00、01、02 となります。残りの2つはまだ消さないでください。最初に、ATCファイルとExemplarファイルを修正する必要があります。そうしないとReader がクラッシュします。

ここで、作成した新しい.FSHファイルを挿入するため、既存の.FSHファイルを削除します（.FSHファイルの上で右クリックし、REMOVE FILEを選択します）。削除が済んだら、空白の部分で右クリックし、INSERT & COMPRESS FILEを選択します。そして、新しい.FSH（例ではsign.fsh）を参照し、挿入します。

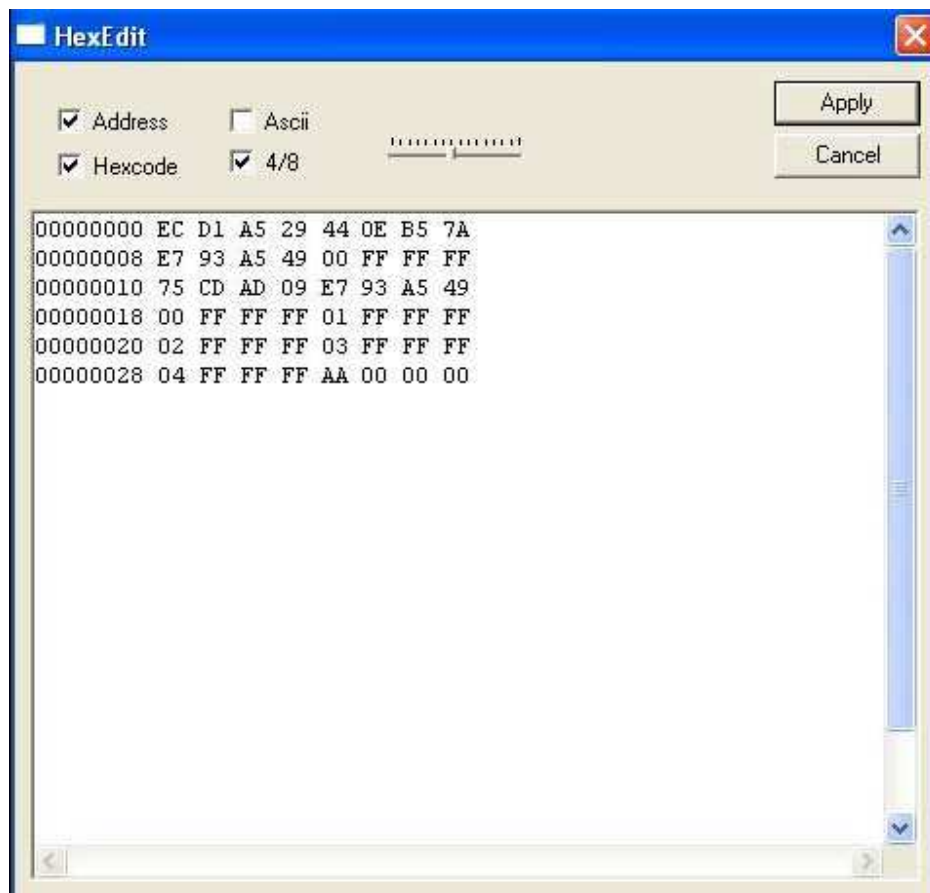
次にInstance IDの末尾が00のAVPファイルを選択し、ツールバーのFILE INFOをクリックします。そして、ポップアップしたダイアログのEDITボタンを押します。



Address (mem)オプションが選択されていることを確認し、Group値とInstance値を、挿入したFSHファイルと同じ値にしてOKボタンを押します。変更が直ちに表に反映されていませんが、次にこのファイルに戻ってきたときには、TGIは変更されています。

同様の操作を他のAVPファイルに対しても行ないます。また、下位バイトの01または02は、この例にならって保持してください。次に、ATCファイルについても同様の操作をしてください。ファイルを選択したときにエラーメッセージが出るかもしれませんが、ここでは無視してかまいません。このメッセージは、開こうとするファイルのAVP参照が、まだ正しくないため表示されるものです。

ATCファイルの上で右クリックし、HEX EDITOR (raw file)を選択します。ダイアログが開いたら、Asciiのチェックを外します。



最初の4バイトはATCファイルのType値を示しています（バイト列は下位から上位に並んでいます）。次の4バイトはFSHファイルのType値を示しています（これらの8バイトは既に訂正されているので、そのままにして手をつけません）。

2行目の最初の4バイトはFSHファイルのGroup値です。ここは新しいFSHファイルのGroup値と置き換えます（下位バイトから … を忘れないように！）。

その次の4バイトはFSHファイルのInstance値です。ここも新しいInstance値と置き換えます（この例では 00 00 FF EE と入力します）。

3行目はAVPファイルのType値（最初の4バイト）とGroup値（次の4バイト）です。Type値はそのままにして、Group値は新しいAVPやFSHファイルに合わせます（全てのファイルは同じGroup値を持つことを忘れないように）。

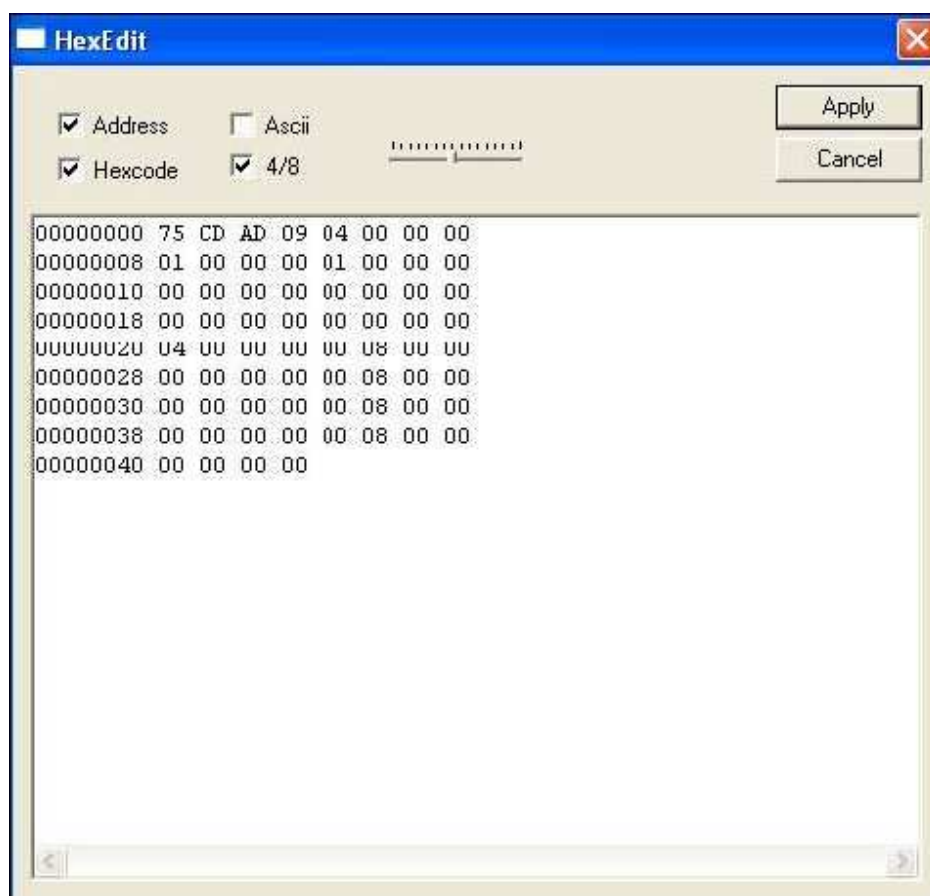
次行からの4バイト組の5セット（4、5行と6行目の最初の4バイト）は、それぞれズームレベルを指しています。ここではズーム1とズーム2は使用しないので、4行目の値は全て00

にします。5行目はズーム3とズーム4を指しているので、これらを各ズームのAVPのInstance値と置き換えます（この例では 00 00 FF EE と 01 00 FF EE となります）。6行目の最初の4バイトはズーム5なので、この例では 02 00 FF EE と記入します。

最後の4バイトは1ズームと1回転視点毎に何コマのフレームを使っているかを示しています。この例では10フレーム（10は16進で0A）なので、0A 00 00 00（下位バイトが先！）とします。

APPLYボタンを押し、ダイアログを閉じます。

AVPファイルのズーム3（この例では EEFF0000）を選択し、16進エディタを開きます。

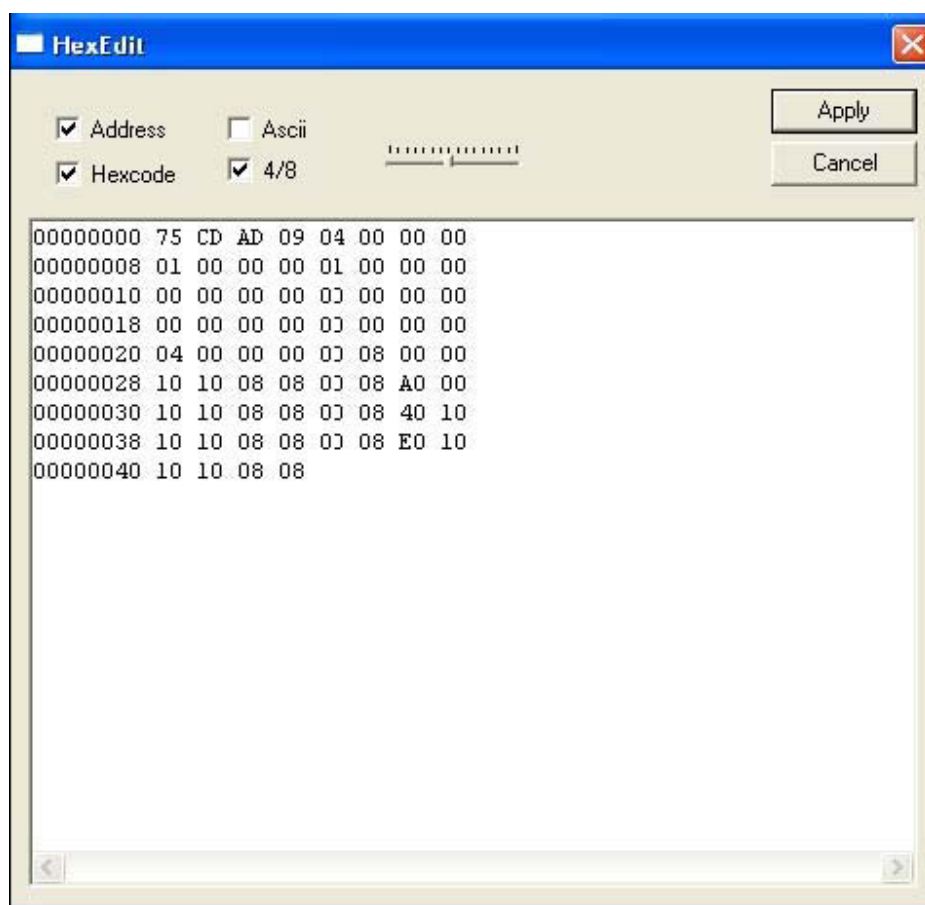


再び下位バイト順の並びで、最初の4バイトはAVPファイルのType値を示しています。これは編集する必要はありません。次の4バイトは5行目の最初の4バイトと共に、各ズームがいくつの回転視点を持っているかを示し、4つ（南東北西）に変わりはないので、そのままにします（指南書末尾の付録に、これについての推論を書いています … 次のプロジェクトではこの推論をテストします）。2、3、4行目はファイルの修正記録と未使用データなので、これらは変更しません。

残りの8バイトづつ4セットは、このズームでの回転視点のパラメータであり、南、東、北、西の順に並んでいます（なぜ、BATのレンダリングがこの順番なのか、なぜ、取出したり、作ったビットマップがこの順番なのかは、ここに理由があります）。

最初のバイトはPLANE、二番目のバイトは 08 のままにします（この数値はフレーム保管データの型でSC4はFSHを参照するの8しか使っていません）。三番目と四番目のバイトはFILEPOSの下位と上位バイトの値です。五番目と六番目はフレームの幅と高さ、七番目と八番目はオフセットのXとYです … 二番目のバイト以外はノートパッドで16進に変換した値から参照できます。【コメント04】

さあ、最初のズームにこれらの数値を詰め込みましょう。終わると、下のようになります：



APPLYボタンを押し、変更を保存します。

次に、ズーム4を16進エディタで開きます（この例ではEEFF0001）。そして、16進に変換した二番目の数値セットを書込みます。続けてズーム5にも最後の数値セットを書込みます。

さて、修正していないファイルが1つ残っています。**Exemplar**ファイルです。

Exemplarの**Instance**はデフォルトにセットしてあるので、この値を独自の値にします（ここではEEFE0000）。変更の仕方は、**FSH**ファイルと**AVP**ファイルの変更を参照してください。

Exemplarファイルをクリックすると、右手のパネルにプロパティが表示されます。

Tweaker				
Name	Name value	Data type	Rep	Value
ParentCohort				0x00000000,0x00000000,0x00000000
Exemplar Type	0x00000010	Lint32	0	Prop
Exemplar Name	0x00000020	String	0	D2010_4x1x1_Animconstructpack_FFFF
AppearanceZoomsFlag	0x0ABFC024	Lint32	0	0x0000001F
Occupant Size	0x27812810	Float32	3	1,1,1
Resource Key Type 4	0x27812824	Lint32	16	0x00000000,0x00000000,0x00000000,0x00000000,0x...
Requester Satisfaction	0x49A1E05A	Lint32	0	Wealth
Prop Time of Day	0x4A149631	Float32	2	0,0
Prop occupant class ID	0xA977A86B	Lint32	0	0x2977AA47
kSC4BuildingModelClassProperty	0xEA123CEF	Lint32	1	0xAA123BF9

アニメに関連するプロパティは、**Exemplar Name**、**Appearance Zoom Flags**、**Occupant Size**、**Resource Key 4** のみです。

プロップの名前は自由に付けてかまいません。著者は**SC4**の命名規則を取り入れた規則に従って命名しています。

タイプ説明__サイズXxYxZ__名前__**exemplar Instance**の高位バイト（タイプ説明は作者の固有の作であること示す説明です）

Exemplar Nameをダブルクリックし、ダイアログに名前を記入します（著者は**D2010_4x4x3_Sign_EEFE**としています）。名前の横の**APPLY**ボタンを押し、次にダイアログの下**APPLY**ボタンを押して確定します。

次に**Appearance Zoom Flags**をセットします。この値は**SC4**にアニメーションが、どのズームで表示されるかを伝えます。値はビットスイッチで、各ビットが1つのズームに対応し、1が **ON**、0が **OFF** となります。

BIT	16	8	4	2	1	
ZOOM	5	4	3	2	1	
Sign.dat	16	8	4	0	0	= 28(10) = 1C(hex)

ここではズーム3、4、5を使っているなので、それら（**ON**にされた）ビット値の合計は上記のように $16+8+8 = 28$ となり、28は16進数では1Cとなります。**Appearance Zoom Flags**をダブルクリックし、1Cをプロパティにセットします。

Occupant sizeはBATで作成したプロップのメーターサイズ（例では4×4×3）です。このプロパティは幅、高さ、奥行き順に構成されていることを覚えておきましょう。サインプロップの場合は 4、3、4 となります。この値でプロパティを変更してください。

【コメント05】

最後に、**Resource Key 4**を変更します。**Resource Key 4**はATCを指しています。**SC4**はこのポインタでアニメーションを見つけることができます。このプロパティをダブルクリックして、ダイアログを出します。リストの中で注目するのは、**rep6**、**rep7**、**rep8** です。これらは順にATCファイルの、**Type**値、**Group**値、**Instance**値 となります。これらの値を変更してください。

ここで、不要となった2つの（AVPファイルの編集の所で、まだ消すなと言った）AVPファイルを削除してください。

さあ、ファイナルステップです。左側の窓の空白で右クリックし、メニューから **SUBFILES and REBUILD DIRECTORY FILE** を選択します。

.DATファイルを保存しましょう。

（.DATファイルを閉じてから、それを再び開き、ATCファイルを選択すると、アニメーションを見ることができます）

【糸冬！】 .DATをpluginsフォルダに入れたら、ロットにアニメをぶち込みましょう。

付 録

1. AVP ファイルの回転視点バイト数

上で完成したサンプルで分かるように、ゲームはAVPファイル内の2バイトを、ズーム毎の回転視点セット数を示すために使っています。ゲームでは4つの回転視点（南、東、北、西）しか使っていないのに、なぜこれを修正する必要があるのか疑問に思われるでしょう。

SC1.DATを調査したところ（そこにアニメーションAVPとATCファイルが格納されています … FSHファイルはSC2.DATに格納されてます）、ズーム当たり8回転視点のAVPに出会いました。

BATでモデルを作るとき、BATはズームごとに4回転視点のレンダリングをします。照明やnite windowsを加えると1回転視点当たり2回のレンダリングが行なわれます。まだテストはしていませんが、8回転視点は4つが日中、4つが夜間のビューであると信じています。

[【コメント06】](#)

2. 影の作成

BATでモデリングしているとき、ベースの四角形を透明にせず、無地の明るい色（ただし、白や灰色以外）にし、ライトをセットしてレンダリングすると影を作ることができます。

ただ、ベースの四角形が表示されてしまうので、新しいアルファチャンネルを作るか、修正する必要が生じるでしょう。しかし、ベースの上に影ができるので、それをビットマップに取り込めばリアルさを高めることができます。

3. S3D アニメーション

SC4はもう一つのタイプのアニメーションプロップを使用しています。それらはS3Dファイルをベースにしています。animconstructpack.dat の exemplar を見る如果能够なら、Resource Key 4 プロパティを開いてみてください。

Reps 6、7、8 がATCファイルのType、Group、Instance を指すことは知っていますが、Reps 14 と15 はS3D型のアニメーションプロップグループをさしています。

Rep 6 と Rep 14 を交換し、Rep 7 と Rep 15も交換すると、独自のS3Dアニメの作成につながるのではないかと想像します。Exemplarにはそれらを付加してあります。

もっと簡単な解決法はないのか？フォーラムでのちょっとした会話から出てきたのですが、意見はこのアプローチにかかわる他の困難に傾いています。

しかし、SC4のdatファイルにはアニメプロップが存在し、それにはexemplarファイルもS3Dファイルも付いてます … 思考の餌ですかね？

[Original by denham2010 in 2005, JP translation by OverPoP in 2007]

■過密市長コメント■

【市長挨拶】

えー、町内会の皆様、本日は会合にご参加いただき真にありがとうございます。このたびの停電並びに断水につきましては隣接市のバカ市長が…（略）

【コメント01】 --- P.2

四角形をLODS調整用に使用していますが、既存のLODSを削除し、一般図形のBOXからリネームでLODSを作成するほうが簡単でしょう。

高さを1/4以下にするよう指示がありますが、これはエクスポート時にBITMAPの大きさが正方形を維持し、また、方位ごとに変化しないようにするためだと思います。後の手作業を考えると必要なことですが、アニメの形状に大きな制約となります。

【コメント02】 --- P.9

FSHファイルはヘッダーとDXT1、DXT3形式の画像データからなっています。DXT1、DXT3形式とBMP形式の画像変換ソースは公開されています。

SC4ModelファイルはDBPF形式で作成され、FSHファイルは通常圧縮された状態で格納されています。理由は分かりませんが、圧縮されないFSHファイルが生成されることもあります。

DBPF形式はSC4の標準形式で、SC4LOTやDATもこの形式で保存されています。DBPF形式の解凍方法は公開されています。

【コメント03】 --- P.13

ここを自動化しないと死にます。

1フレームのビットマップサイズは、AVPファイルにビットマップの組み合わせを指定する項目がないので、256×256が最大ようです。

SC4がゲームで使用するS3Dファイルには、ここで使用されているようなアルファは存在しません（夜景の照明で明るい部分を抜き出すアルファは存在します）。BATがモデルをエクスポートする際に、テクスチャに真黒（R,G,B=0）を使わず、モデル周辺のみ真黒を使用しているのであれば、わざわざアルファ用のSC4Modelファイルを作らなくても、単純なフィルタでアルファを作成できそうです。

【コメント04】 --- P.20

オフセットで示す点が何を指しているのかよくわかりません。希望的には、モデルの原点（x,y,z=0）またはLODの3軸中心であって欲しいところです。

SC4の画面はアイソメ図法（火星あたりから超！超！望遠カメラで地球の都市を見た感じ）で描画されています。この図法だと物体が視線方向に移動しても、画面上では位置や大きさが変化しません。そのため、ロット上のプロップ設置位置にビットマップのオフセットを合わせるだけで、3Dを意識しなくてもアニメは正しく表示されるのではないかと想像します。アニメの画像も正方形にこだわる必要はなくなると思います。

ただし、BATがエクスポートした画像は通常長方形で、しかも、画像の中心は原点やLODの中心とは合っていません。アニメのビットマップとは別にオフセット位置を知らせる画像が必要になると思います。

【コメント05】 --- P.22

アニメのデータにはLODSがないので、他のオブジェクトとの前後関係を知る情報は、ロット上の設置位置とOccupant sizeだけのようです。

【コメント06】 --- P.23

これが解決できない場合、夜景はSC4Model内のFSHの昼間と夜間と照明用アルファを合成して、1枚のビットマップを作成し、それでフレームを作る必要があります。また、プロップは昼間用と夜間用に分け、タイマーで表示をコントロールしなければなりません。

【コメント番外】

うーん、こうやってコメントを付けると、自分もMODの大家に見える（^；