# mod_SC3D_E 説明

*Index*


- 他のモジュールの推奨
- 一言説明
- 定義の継承
- 依存するモジュール
- 3DO 形式
- 3DO 形式データのサンプル
- 命令リファレンス


## ＜ 他のモジュールの推奨 ＞

通常はこのモジュールではなく「mod_SC3D_E2」を使用することをお勧めします。

「mod_SC3D_E2」がサポートする B3DO 形式では 3DO 形式のデータに比べて保存できる情報が制限されますが、それらの情報はそもそも数学的に意味を持たないエクストラ情報なので、通常は問題ありません。
そして、B3DO 形式のデータの方が、オブジェクト構造体との相互変換処理にかかる時間が圧倒的に少ないです。

## ＜ 一言説明 ＞

mod_SC3D_C で使用するオブジェクト構造体のファイルの入出力のフォーマットを定義します。
フォーマット変換ルーチンを提供します。

## ＜ 定義の継承 ＞

mod_SC3D の A,B,C ユニットで定義されたことは本ユニットでも引き続き有効となります。

## ＜ 依存するモジュール ＞

mod_CharStr
mod_fx3

上記のモジュールに依存します。必ずこのモジュールを使用可能な状態にしておいてください。

## ＜ 3DO 形式 ＞

これは、mod_SC3D_C で使用するオブジェクト構造体をテキストデータとして表現したものです。
このデータからオブジェクト構造体に変換することができます。

スクリプト内で毎度毎度オブジェクトを作成することなく、ファイルから読み込んでスクリプトの冗長化を抑えることができるというメリットがあります。

3DO 形式にはいくつか制約がありますが、一つここで述べておくとすれば、1 行あたり 4096 バイトまでが認識の限界ということです。通常はこれほど長くならないので注意する必要はありません。

以下で、3DO 形式について説明します。

＜ 文字の判別について ＞

#START 等の「トークン」では、大文字・小文字の区別はされません。どちらで書いても問題ありません。

＜ コメントについて ＞

//の右側は行末までがコメントとして扱われます。

＜ Tab について ＞

任意の位置に Tab を入れることができます。

＜ パラメータについて ＞

パラメータ同士は,で区切ってください。
パラメータと,の間に半角スペースを挟んでもかまいません。

数値パラメータについて、実数型・整数型の区別は本モジュール側で適宜行うので、気にする必要はありません。

また、パラメータとして数式を使うことができます。
例えば、「20+15」、「15 + (20-3)*6」、「20*sin(m_pi/4)」等です。
c_pi と c_e はそれぞれ円周率,ネイピア数として扱われます。
書式は本モジュールとは別の「mod_fx」というモジュールのものと同じです。HSP で数式を記述する時とほとんど変わりません。但し、変数は使えません。
数式を扱う機能は「mod_fx」により実現されていますので、挙動もこれに従います。

＜ 3DO 形式のデータであることの宣言 ＞

テキストデータの先頭に必ず
#CLASS 3DObjectData
と記述してください。
これは「3DO 形式のデータだ」と宣言する意味を持ちます。

＜ 3DO データの始まりと終わりの宣言 ＞

#START 3DO

で始まる行と、
#END 3DO
で始まる行に挟まれた部分が3DOデータとみなされます。
このエリアの外の文字列は無視されます。
挟まれた領域を1ファイル中に複数個作っても、認識されるのは最初の1エリアのみです。

＜ オブジェクトの始まりと終わりの宣言 ＞

#START Obj
で始まる行と、
#END Obj
で始まる行に挟まれた部分が1つのオブジェクトとみなされます。
この部分を「オブジェクト部分」と呼びます。
挟まれた領域を1ファイル中に複数個作っても、認識されるのは最初の1エリアのみです。

＜ オブジェクト部分内の構造 ＞

#NUM_PARTS
でオブジェクトが保有するパーツの数を指定します。 (例) #NUM_PARTS 8

#DrawPriorityClass
で描画順序優先度クラス(1～10)を指定します。 (例) #DrawPriorityClass 2

#POS
でオブジェクトの中心のグローバル座標を指定します。 (例) #POS 1.5 ,2 ,3

#->e_obj_x
#->e_obj_y
#->e_obj_z
で→e_obj_x ,→e_obj_y ,→e_obj_zのx,y,z成分を指定します。
(例)

```
#->e_obj_x 1 ,0 ,0
#->e_obj_y 0 ,1 ,0
#->e_obj_z 0 ,0 ,1
```

ただし、3DO形式からオブジェクト構造体への変換後にmod_SC3D_Cモジュールの「MSC3D_C_SetObj」命令により姿勢を決定するのが普通なので、通常は→e_obj_x,y,zの値は適当な値でかまいません。通常は3DO形式の状態であらかじめ姿勢を決定しておくことにあまり意味はありません。

#SWITCH_ZSORT
はパーツのZソートスイッチです。(例) #SWITCH_ZSORT 1

＜ パーツの始まりと終わりの宣言 ＞

オブジェクト部分内で、
#START Part
で始まる行と
#END Part
で始まる行に挟まれた部分が一つのパーツと見なされます。
この部分を「パーツ部分」と呼びます。
このパーツ部分はいくつあってもよく、最初のものから順にパーツのID(0,1,2,･･･)となります。

＜ パーツ部分内の構造 ＞

#NAME
でパーツの名前を指定します。（例）#NAME ″尾翼1″
このデータは幾何学的に意味を持たないので、#NAME 自体を省略してもかまいません(つまり名無し状態)。
名前の両端は必ず″″で囲んでください。

#EXPLAIN
でパーツの説明を1022バイト以内で記述します。(例) #EXPLAIN ″フロント硝子/%α50くらいがいいか...?″
/% で改行を表します。従って、/% という文字列は説明に使えませんので注意してください。
このデータは幾何学的に意味を持たないので、#EXPLAIN 自体を省略してもかまいません。
説明の両端は必ず″″で囲んでください。

#NAME と#EXPLAIN は「エクストラデータ」に分類されます。

#CLASS
でパーツの種類を指定します。
dot で「点」
line で「線分」
square で「四角形平面」
sphere で「球」
を表します。（例）#CLASS ″square″

#VISIBLE
でパーツの可視/不可視を設定します。
(0,1) = (不可視,可視)
(例) #VISIBLE 1

以降の記述はパーツの種類により異なります。以下を御覧ください。

＜ dot の場合 ＞

#POS_ObjCoordinate
でオブジェクト座標系座標を指定します。(例) #POS_ObjCoordinate 1.2 ,0 ,1

#RADIUS
で描画時の半径〔px〕を指定します。（例）#RADIUS 3

#BLENDRATE
でブレンド率を指定します。（例）#BRANDERATE 128

#SWITCH_CONSIDER_LIGHT
は光の効果の適用スイッチです。 (0,1) = (適用しない,する)
(例) #SWITCH_CONSIDER_LIGHT 1

#MINLIGHT
で最低光量を指定します。（例）#MINLIGHT 0.2

#SWITCH_PICTEXTURE
は画像割り当てのスイッチです。 (0,1) = (無,有)
(例) #SWITCH_PICTEXTURE 0

#COLOR

で、画像を割り当てない場合の、点の色のRGB値を指定します。
(例) #COLOR 255,128,255

#ID_WND_PICTEXTURE
画像を割り当てる場合は、これで素材画像の参照先のウィンドウのIDを指定します。
(例) #ID_WND_PICTEXTURE 1

#POS_A_PICTEXTURE
画像を割り当てる場合は、これで素材画像の左上の点のウィンドウ座標系座標を指定します。
(例) #POS_A_PICTEXTURE 0,20

#SIZE_PICTEXTURE
画像を割り当てる場合は、これで素材画像のx,yサイズを指定します。
(例) #SIZE_PICTEXTURE 64,32

#SWITCH_TRANSPARENT_PICTEXTURE
画像を割り当てる場合は、これで透過色の有効,無効を指定します。
(0,1) = (無効,有効) (例) #SWITCH_TRANSPARENT_PICTEXTURE 0

#COLOR_TRANSPARENT_PICTEXTURE
画像を割り当て、透過色が有効な場合は、これで透過色のRGB値を指定します。
(例) #COLOR_TRANSPARENT_PICTEXTURE 255,0,0

#GLOSSRATE
で光沢度(0～255)を指定します。 (例) #GLOSSRATE 128

< lineの場合 >

#POS_A_ObjCoordinate
#POS_B_ObjCoordinate
で2端点のオブジェクト座標系座標を指定します。
(例)
#POS_A_ObjCoordinate 1,1,1
#POS_B_ObjCoordinate -1,-1,-1

#THICKNESS
で描画時の太さ〔px〕を指定します。 (例) #THICKNESS 2

#COLOR
で色のRGB値を指定します。 (例) #COLOR 255,128,255

#BLENDRATE
でブレンド率を指定します。 (例) #BRANDERATE 128

#SWITCH_CONSIDER_LIGHT
は光の効果の適用スイッチです。 (0,1) = (適用しない,する)
(例) #SWITCH_CONSIDER_LIGHT 1

#MINLIGHT
で最低光量を指定します。 (例) #MINLIGHT 0.2

#GLOSSRATE
で光沢度(0～255)を指定します。 (例) #GLOSSRATE 128

＜ square の場合 ＞

#POS_A_ObjCoordinate
#POS_B_ObjCoordinate
#POS_C_ObjCoordinate
#POS_D_ObjCoordinate
で４点A,B,C,Dのオブジェクト座標系座標を指定します。
※A,B,C,Dは環状に。
(例)
#POS_A_ObjCoordinate 1,1,1
#POS_B_ObjCoordinate -1,1,1
#POS_C_ObjCoordinate -1,-1,1
#POS_D_ObjCoordinate 1,-1,1

#BLENDRATE
でブレンド率を指定します。（例）#BRANDERATE 128

#SWITCH_CONSIDER_LIGHT
は光の効果の適用スイッチです。（0,1）=（適用しない,する）
（例）#SWITCH_CONSIDER_LIGHT 1

#MINLIGHT
で最低光量を指定します。（例）#MINLIGHT 0.2

#DRAWMODE
で描画モードを指定します。
(0,1,2) =（ワイヤーフレーム,単色塗り潰し,テクスチャ張付け）
（例）#DRAWMODE 1

#COLOR
で、ワイヤーフレームまたは単色塗り潰しの場合の、色のRGB値を指定します。
（例）#COLOR 255,128,255

#THICKNESS
で、ワイヤーフレームの場合の描画時の太さ〔px〕を指定します。
（例）#THICKNESS 2

#ID_WND_PICTEXTURE
テクスチャ張付けの場合は、これで素材画像の参照先のウィンドウのIDを指定します。（例）#ID_WND_PICTEXTURE 1

#GOB_POSX_PICTEXTURE
#GOB_POSY_PICTEXTURE
テクスチャ張付けの場合は、これで素材画像の頂点A,B,C,Dのx,y座標群を指定します。
HSPのgsquare命令のパラメータと同じ要領です。
※A,B,C,Dは環状に。
(例)
#GOB_POSX_PICTEXTURE 0,99,99,0
#GOB_POSY_PICTEXTURE 0,0,199,199

#SWITCH_TRANSPARENT_PICTEXTURE
テクスチャ張付けの場合は、これで透過色の有効,無効を指定します。
(0,1) =（無効,有効）（例）#SWITCH_TRANSPARENT_PICTEXTURE 0

#COLOR_TRANSPARENT_PICTEXTURE
テクスチャ張付けで透過色が有効な場合は、これで透過色の RGB 値を指定します。
(例) #COLOR_TRANSPARENT_PICTEXTURE 255,0,0

#GLOSSRATE
で光沢度(0～255)を指定します。 (例) #GLOSSRATE 128

＜ sphere の場合 ＞

#POS_ObjCoordinate
で中心のオブジェクト座標系座標を指定します。
(例) #POS_ObjCoordinate 1.2 ,0 ,1

#OutsideDiameter
で外径を指定します。 (例) #OutsideDiameter 10

#COLOR
で色の RGB 値を指定します。 (例) #COLOR 255,128,255

#BLENDRATE
でブレンド率を指定します。 (例) #BRANDERATE 128

#SWITCH_SHELL
で中空構造なのか、充填構造なのかを指定します。(0,1) = (充填,中空)
(例) #SWITCH_SHELL 0

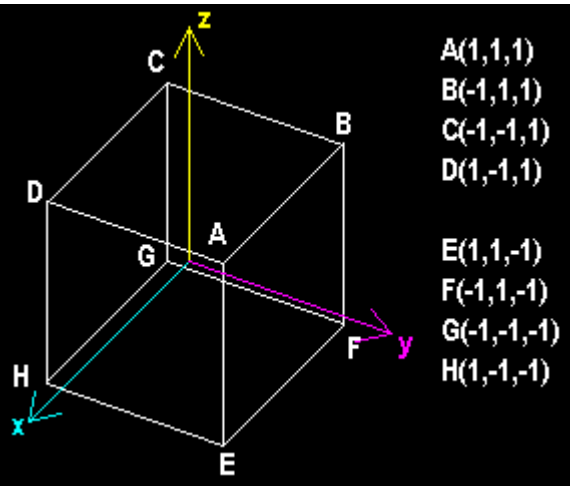
#CALIBER
中空構造の場合は、これで内径を指定します。 (例) #CALIBER 8

#SWITCH_CONSIDER_LIGHT
は光の効果の適用スイッチです。 (0,1) = (適用しない,する)
(例) #SWITCH_CONSIDER_LIGHT 1

#MINLIGHT
で最低光量を指定します。 (例) #MINLIGHT 0.2

## ＜ 3DO 形式データのサンプル ＞

3DO のフォーマットに従って下の図のような立方体を記述した場合のサンプルを下に示します。




```
+++++++++++++++++++++++++++++++++ この下から +++++++++++++++++++++++++++++++++
#CLASS 3DObjectData
//立方体サンプル
#START 3DO
	#START Obj
		#NUM_PARTS 6
		#DrawPriorityClass 1
		#POS 0,0,0
		#->e_obj_x 1,0,0
		#->e_obj_y 0,1,0
		#->e_obj_z 0,0,1
		#SWITCH_ZSORT 1

		#START Part	//面ABCD
			#NAME "面ABCD"
			#EXPLAIN "1辺の長さは2。/%赤色。"
			#CLASS "square"
			#VISIBLE 1
			#POS_A_ObjCoordinate 1,1,1
			#POS_B_ObjCoordinate -1,1,1
			#POS_C_ObjCoordinate -1,-1,1
			#POS_D_ObjCoordinate 1,-1,1
			#BLENDRATE 255//ブレンド率max
			#SWITCH_CONSIDER_LIGHT 1	//光の効果を適用
			#MINLIGHT 0.2
			#DRAWMODE 1	//単色塗り潰し
			#COLOR 255,0,0//赤色
			#GLOSSRATE 128//光沢度128
		#END Part

		#START Part	//面EFGH
			#NAME "面EFGH"
			#CLASS "square"
			#VISIBLE 1
			#POS_A_ObjCoordinate 1,1,-1
			#POS_B_ObjCoordinate -1,1,-1
			#POS_C_ObjCoordinate -1,-1,-1
			#POS_D_ObjCoordinate 1,-1,-1
			#BLENDRATE 255
			#SWITCH_CONSIDER_LIGHT 1
			#MINLIGHT 0.2
			#DRAWMODE 1
```

```
        #COLOR 255,0,0
        #GLOSSRATE 128
#END Part

#START Part    //面ABFE
        #NAME "面ABFE"
        #CLASS "square"
        #VISIBLE 1
        #POS_A_ObjCoordinate 1,1,1
        #POS_B_ObjCoordinate -1,1,1
        #POS_C_ObjCoordinate -1,1,-1
        #POS_D_ObjCoordinate 1,1,-1
        #BLENDRATE 255
        #SWITCH_CONSIDER_LIGHT 1
        #MINLIGHT 0.2
        #DRAWMODE 1
        #COLOR 255,0,0
        #GLOSSRATE 128
#END Part

#START Part    //面BCGF
        #NAME "面BCGF"
        #CLASS "square"
        #VISIBLE 1
        #POS_A_ObjCoordinate -1,1,1
        #POS_B_ObjCoordinate -1,-1,1
        #POS_C_ObjCoordinate -1,-1,-1
        #POS_D_ObjCoordinate -1,1,-1
        #BLENDRATE 255
        #SWITCH_CONSIDER_LIGHT 1
        #MINLIGHT 0.2
        #DRAWMODE 1
        #COLOR 255,0,0
        #GLOSSRATE 128
#END Part

#START Part    //面CDHG
        #NAME "面CDHG"
        #CLASS "square"
        #VISIBLE 1
        #POS_A_ObjCoordinate -1,-1,1
        #POS_B_ObjCoordinate 1,-1,1
        #POS_C_ObjCoordinate 1,-1,-1
        #POS_D_ObjCoordinate -1,-1,-1
        #BLENDRATE 255
        #SWITCH_CONSIDER_LIGHT 1
        #MINLIGHT 0.2
        #DRAWMODE 1
        #COLOR 255,0,0
        #GLOSSRATE 128
#END Part

#START Part    //面DAEH
        #NAME "面DAEH"
        #CLASS "square"
        #VISIBLE 1
```

```
                        #POS_A_ObjCoordinate 1,-1,1
                        #POS_B_ObjCoordinate 1,1,1
                        #POS_C_ObjCoordinate 1,1,-1
                        #POS_D_ObjCoordinate 1,-1,-1
                        #BLENDRATE 255
                        #SWITCH_CONSIDER_LIGHT 1
                        #MINLIGHT 0.2
                        #DRAWMODE 1
                        #COLOR 255,0,0
                        #GLOSSRATE 128
                #END Part

        #END Obj
#END 3DO
++++++++++++++++++++++++++++++++++ この上まで ++++++++++++++++++++++++++++++++++
```

## ＜ 命令リファレンス ＞

***Index***


- MSC3D_E_init
- MSC3D_E_3DOToObjstruct
- MSC3D_E_GetExtraDataFrom3DO
- MSC3D_E_ObjstructTo3DO


| 命令 | 説明 |
|---|---|
| MSC3D_E_init | モジュールを初期化します。<br>本モジュールを初めて使う前にこの命令を必ず実行してください。<br><br>[書式]<br><br>MSC3D_E_init |
| MSC3D_E_3DOToObjstruct | 3DO 形式のデータから、mod_SC3D_C で使えるオブジェクト構造体のデータを作成します。<br><br>[書式]<br><br>MSC3D_E_3DOToObjstruct str ,array ,buf ,opt ,label<br><br>str ： 3DO 形式のデータが格納された文字列データ(変数名可)<br>array ： オブジェクト構造体の生成先変数(初期化不要)<br>buf ： エラーメッセージを格納する変数(初期化不要)<br>opt ： エクストラデータ(パーツの名前と説明)を確認するかどうかのオプション。(0,1)=(しない,する)<br>後で「MSC3D_E_GetExtraDataFrom3DO」命令を使う場合は 1 にしてください。 |

| | |
|---|---|
| | label ： 処理が1%完了する毎にサブルーチンジャンプするラベル情報を格納したラベル型変数<br><br>label で指定したラベルでは処理の達成度〔%〕が「progress@SC3D_E_30」変数から取得できます。<br><br>無事終了した場合は stat の値が 0 になります。<br>エラーが発生した場合はエラーが発生した行番号(1,2,3,･･･)が stat に格納され、buf にエラーメッセージが日本語文字列で保存されます。<br>但し、エラー発生行が特定できない場合は stat の値が-1 になります。 |
| MSC3D_E_GetExtraDataFrom3DO | 3DO 形式のデータから、パーツの名前と説明を取得します。<br><br>最後に実行された「MSC3D_E_3doToObjstruct」命令で扱われた 3DO データが対象となります。<br><br>[ 書式 ]<br><br>MSC3D_E_GetExtraDataFrom3DO array1<br><br>array1 ： 結果を格納する配列変数(初期化不要)<br><br>array1 は文字列型配列です。<br>要素(m,0),(m,1) (※m = 0,1,2,･･･)<br>にそれぞれ、m 番目のパーツの 名前,説明 が格納されます。<br><br>無事終了した場合は stat の値が 0 になり、それ以外の場合は 1 になります。 |
| MSC3D_E_ObjstructTo3DO | オブジェクト構造体から、3DO 形式のデータを作成します。<br><br>[書式]<br><br>MSC3D_E_ObjstructTo3DO var ,opt ,array1 ,array2<br><br>var ： 3DO 形式のデータの格納先変数(初期化不要)<br>opt ： エクストラデータの書き込みオプション(0,1)=(書き込まない,書き込む)<br>array1 ： オブジェクト構造体<br>array2 ： エクストラデータを格納した配列変数<br><br>array2 は文字列型配列変数です。<br>opt を 0 にした場合は適当なダミー変数を指定してかまいません。<br>要素(m,0),(m,1) (※m = 0,1,2,･･･)<br>にそれぞれ、m 番目のパーツの 名前,説明 を格納してください。<br><br>無事終了した場合は stat の値が 0 になり、エラーが発生した場合は 1 になります。 |