



一般的なCGソフトでは、モデラーによって3Dオブジェクトを作りますが、TouchDesignerでは基本形からノードを繋げることで加工していきます。

奇妙な作り方のようですが、実はそれは「作る過程」の履歴を残しているに過ぎません。しかも途中過程の一部を後から修正・変更できる(「その後に行う加工」は維持したまま!)という特長があります。

残念ながら、複雑なモデリングには向いていません。複雑なモデルを使いたい場合は別ソフトで作って、インポートします。Obj形式を使います。UVマップもインポート可能です。

分類		OP名	説明	重要度
新規作成系	線	Line SOP	直線生成	
		Inverse Curve SOP	Inverse Curve CHOPから生成	
	平面	Rectangle SOP	四角形生成	
		Circle SOP	円生成	
		Grid SOP	グリッド面生成	★★★
		Font SOP	文字生成	★
	立体	Box SOP	ボックス生成	★
		Sphere SOP	球体生成	★★★
		Limit SOP	CHOPを元に形状作成	
		Iso Surface SOP	ISOサーフェース生成	
		Tube SOP	筒生成	
		Torus SOP	トーラス(ドーナツ状物体)生成	★
		Superquad SOP	スーパークアド生成	
		Metaball SOP	メタボール生成	★
		Arm SOP	アーム(曲げジョイント付きの筒)生成	
		LSystem SOP	LSystemの枝状分岐図形生成	☆
		Sprite SOP	CHOPのXYZ指定で常にカメラ方向を向くスプライトを作る	☆
		外部から	File SOP	形状ファイル読み込み
Object Merge SOP	外部コンポーネントからパス指定でオブジェクト持込			
DAT to SOP	DATデータをSOP化			
Trace SOP	2Dイメージをトレース。この後、Extrude SOPすることが多い		☆☆	
モデリング系	大まか・汎用	Point SOP	ポイント単位のエクプレッション操作	★★★
		Primitive SOP	プリミティブの操作。PointSOPのプリミティブ版。	
		Transform SOP	移動・拡大縮小	★★★
	点/線単位	Capture Region SOP	範囲による点選択	
		Capture SOP	点選択	
		Group SOP	ポイントのグループ化	★★
		Sort SOP	ポイントのソート	★
		Add SOP	ポイントを増やす	★
		Delete SOP	ポイントの削除	★
		Align SOP	位置揃え	
		Deform SOP	ポイントベースの形状変形	
		Basis SOP	スプラインカーブの加工	
	面/立体単位	Cap SOP	穴を塞ぐ	
		Clip SOP	クリッピング	
		Extrude SOP	引き出し加工。ベベル(面取り)にも使う。	★
		Hole SOP	穴開け加工	
		Revolve SOP	回転体	★
		Sweep SOP	掃引	☆
		Divide SOP	分割	
		Bridge SOP	ロフト?的に包む	
		Trim SOP	トリムして見えなくする	
		Carve SOP	スライス・穴あけなど	
		Line Thick SOP	カーブから布を作る	
		オブジェクトごと変形	Noise SOP	各点をノイズ的に動かす
	Twist SOP		捻じ曲げ加工	★★
	Vertex SOP		vertex(ポイント)単位でうのように動かす	
	Clay SOP		ねんどの変形加工	
	Magnet SOP			
	Lattice SOP		ラティス変形(変形コントロールするオブジェクトを使った変形)	
	Spring SOP		ポリゴンのエッジや各ポイントに付加された質量を元に、スプリング(ばね)変形	
	Fractal SOP		フラクタル加工	
	Creep SOP		PathなSOPに応じて入力SOPを変形する	
	ポリゴン加工	Polyloft SOP	ロフト	

		Polypatch SOP		
		Polyspline SOP		
		Polystitch SOP	ポリゴンを縫い合わせる。異なる物体として面を評価されることに起因する割れ	
複数オブジェクト合成		Boolean SOP	ブーリアン演算加工	★★
		Surfsect SOP	ブーリアン演算加工 (NURBS/Bezier形状用)	
		Blend SOP	同じポイント構成を持つ2物体を比率でブレンド(顔モデル等で使われるブレンド)	
		Sequence Blend SOP	似たポイント構成を持つ複数物体を比率でブレンド。	
		Fillet SOP	2つのオブジェクトをモーフィング合成	
		Project SOP	写像(プロジェクション)加工	
		Ray SOP	プロジェクションの一種	
		Rails SOP	2オブジェクト間の掃引	
		Join SOP	サーフェース間接続	
	パーティクル		Particle SOP	パーティクル発生
		Force SOP	パーティクルに力を加える。Metaballをインプットする	★★
ボーン		Bone Group SOP		
		Joint SOP		
調査中		Curveclay SOP		
		Curvesect SOP		
		Model SOP	モデリングツール?	
		Fit SOP	スプラインカーブ/面を一連のポイント列/メッシュにフィットさせる	
		Profile SOP		
		Skin SOP		
		Stitch SOP	縫合	
	エフェクト系		Trail SOP	残像を作る
		Wireframe SOP	ワイヤーフレーム化加工	☆
アトリビュート操作系		Attribute SOP	属性のリネームと削除	
		Attribute Create SOP	法線とタンジェント属性を付加。バンプマップに必要。	★
		Facet SOP	ファセット(法線を平均化) = スムースシェーディング	★★★
コンバート系		Convert SOP	NURBS/Polygonなど形式コンバート	★★
		Subdivide SOP	サブディビジョンサーフェース化	☆☆
		Polyreduce SOP	ポリゴン減少加工	☆
		Refine SOP	形状をなるべく保ちつつ、頂点数を増減	
		Resample SOP	形状をなるべく保ちつつ、ポリゴン密度を均等になるようにする	
		Tristrip SOP	三角形分割。レンダリングを速くできる。	
マテリアル設定系		Material SOP	マテリアル設定	★★★
		Texture SOP	テクスチャマッピング設定	★★★
フロー制御		Copy SOP	入力されたSOPのコピー。コピー先座標群を示すSOPを2ndインプットに入れるか、コピー先への移動定義+コピー数でコピーできる。「コピー元」に変化を与えた結果をコピーする機能があり、入力元で定義したパラメータやアトリビュートに対して変数\$CYなどを用いて値をセット可能。	★★★
		Merge SOP	複数SOPを1つのSOPにまとめる(だけ)。形状合成的なことはしない。※異なるマテリアルを持つSOPをまとめることはできない	★★★
		Switch SOP	フローのスイッチング。入力元SOPを切り替えて出力する。	★★★
		Cache SOP	入力された図形をキャッシュ(蓄積)する	
		Channel SOP	CHOPから来たチャンネル値を受信	★★
		Select SOP	別のSOP(上階層でも下階層でもない)からの取り込み	★★
		In SOP	属するコンポーネントの外からの入力インターフェースを作る	★
		Out SOP	属するコンポーネントの外への出力インターフェースを作る	★
その他		LOD SOP	カメラから遠い物体を省略型オブジェクトに置き換える	☆
		Null SOP	ヌル(何もしない)	

## Copy SOP

他のSOPのジオメトリをコピーすると同時に、それぞれのコピーに対して加工を施すこともできる。※インプットされたポイント上にコピーすることもできる。

※この「それぞれのコピーに対して加工を施す」機能は、移動・回転などの後加工だけでなく、インプット元、すなわち上流のSOPジオメトリに対して、各コピーごとに異なる加工をしてしまえる手段が用意されているのが大きな特徴！(Stamp Page参照)

## ■パラメータ Copy Page

Source Group /sourceGrp

Template Group /templateGrp

Number of Copies /ncy

Primitives Per Point /nprims

Rotate to Normal /nml

コピーソースのグループ限定

テンプレートのグループ限定

コピー数。テンプレートがインプットされてる場合は不要

各ポイントにいくつプリミティブを置くか

(テンプレート使用時のみ)法線に従ってコピーの向きを変えるか否か。

Transform Cumulative /cum	これを使う際は、上流で(PointSOPなどを用いて)各ポイントに法線アトリビュートが持たされていないならば、当然うまくいかない。
Transform Order /xOrd	各コピーに対するトランスフォームを、累積(累加的)に適用するか否か
Rotate Order /rOrd	回転・移動・スケールの実行の順。違えば結果は変わる。
Translate, Rotate, Scale, Pivot	回転のx,y,z軸の順
Uniform Scale /scale	コピーされるごとに適用される変形量。ローカル変数\$CYなどを使えば各コピー別操作も可能。
Normals Maintain Length /vlength	スケール変形時、xyz軸を同期するか否か
Create Output Groups / Copy Groups /newg	スケール変形時、ベクトル型アトリビュート(normals, velocityなどの)「長さ」はリサイズせずに維持するか
Look At /lookat	本SOPのアウトプットを新たなグループとして定義
Up Vector /upvector[xyz]	(カメラオブジェクトによくあるようなLook At機能) Look At機能の「上」方向を定義

## ■ローカル変数

\$CY	各コピーインスタンスに振られる番号
\$NCY	コピー数。コピーを等間隔配置するとき位置計算ソースに使うと便利。(例:円周配置のときに $rz = 360/\$NCY$ )
\$PT	テンプレート側に入ってきた各ポイントの番号
\$NPT	テンプレート側に入ってきたポイントの数
\$AGE	テンプレートとして入ってきた各パーティクルの秒齢
\$LIFE	テンプレートとして入ってきた各パーティクルの、寿命内の現在の年齢の比率。0(誕生)から1(死)までの値。
\$ID	テンプレートとして入ってきた各パーティクルのID
以下は、テンプレートのインプットに何もつながれていないならば、コピー(される)ジオメトリのインプットの値を返す。	
\$CEX \$CEY \$CEZ	ジオメトリの中心(絶対位置)
\$XMIN \$XMAX \$YMIN \$YMAX \$ZMIN \$ZMAX	ジオメトリの最大範囲(絶対位置)
\$BBX \$BBY \$BBZ	ジオメトリの最大範囲内における相対的なポイント位置
\$SIZEX \$SIZEY \$SIZEZ	ジオメトリの最大範囲のサイズ

## ■パラメータ - Stamp Page

Stampは、コピー元ジオメトリに対して、各コピーごとに異なる加工をしてしまえる機能。

Stamp Inputs	本機能のOn/Off
Param 1-10	インプット元で定義したパラメータ名と、コピーインスタンスごとに異なる値を返す変数(式)を記入

(例)3角形以上の多角形を多数作る

- 1.上流(コピー元)としてCircleSOPを作成。ポリゴンにする。
- 2.«number of Divisions»に「param(「sides”,3)」と記入。  
このparam()は、外から参照・操作可能なパラメータの定義。操作される前のデフォルト値を同時設定できる。  
これで、デフォルト値3の「sides」というパラメータが、Circleの分割数に反映されるようになった。
- 3.下流にCopy SOPを接続し、いくつかのコピーを作るように設定。
- 4.Stamp InputsをOnにして、Param1に「side」Value1に「\$CY+3」と記入。  
これで、コピー数が増えるたびに三角形から多角形が増えていくようなものが作られた。

## Point SOP ポイント単位のエクスペッション操作

ポイントの位置・色・テキストチャコーディネイト・法線などの属性を、操作できます。

また、元のSOPIには用意されていない新たなカスタム属性を作って持たせることもできます。

例えば、ポイントに色を与えたり、法線を反転したり、ポジションX,Y,Zの中でエクスペッションを使用すれば、与えられた入力ポジションから新しい場所へ移動させることもできます。

第2インプットが、第1インプットより少ないポイント数だった場合、第2の入力でのポイントはサイクル的に割り当てられます。

## ■パラメータ Standard Page

Position /tx /ty /tz	元のXYZポジションに加工を施すエクスペッションをここに書き込める。 \$TX,\$TY,\$TZが元の各点のポジションを意味するローカル変数。 例えば、txに\$TX+5と入力すれば、全ての点のXポジションに5が加算される。
Keep / Add / No Weight /weight	各点に、weightアトリビュートを維持・追加、または無くす。 インプットされた元の値を示すローカル変数は\$WEIGHT。値の範囲は0.0001から無限大まで。
Keep / Add / No Point Colors /diffg /diffg /dif	各点に、colorアトリビュートを維持・追加、または無くす。 インプットされた元の値を示すローカル変数は\$CR, \$CG, \$CBです。
Keep / Add / No Alpha /alpha	各点に、アルファ値のアトリビュートを維持・追加、または無くす。 インプットされた元の値を示すローカル変数は\$CAです。値の範囲は0.0(完全透明)から1.0(完全不透明)まで。
Keep / Add / No Normals /nx /ny /nz	各点に、法線のアトリビュートを維持・追加、または無くす。 インプットされた元の値を示すローカル変数は\$NX, \$NY, \$NZです。法線を反転する場合は、-\$NX -\$NY -\$NZと記入すればよい。 Turbulence, Facet, Copy SOPで使われる。

Custom Attrib /custom1	(例)NormalをばらばらにするにはAddNormalで カスタム属性の名前を定義する。
Custom Attrib Size /custom1size	カスタム属性のサイズ(変数型)を定義する。
Value /custom1val	カスタム属性の値を設定する。
Keep / Add / No Texture /mapu /mapv /mapw	各点に、テクスチャのUVマッピングのための定式を維持・追加、または無くす。 インプットされた元の値を示すローカル変数は\$MAPU, \$MAPV, \$MAPW。

#### ■パラメータParticle Page

Keep / Add / No Mass/ Drag /mass /drag	各点のmass・drag属性の維持・追加・削除
Keep / Add / No Tension /tension	Tensionはポイントが接続される端の弾力に影響。
Keep / Add / No Spring Constant /springk	「Spring Constant」はよく知られている物理的属性。
Keep / Add / No Point Velocity /vx /vy /vz	各点のvelocity。XYZ方向にパーティクルの速度量を定義。
Keep / Add / No Up Vector /upx /upy /upz	各点の“上方向”を示す属性。Copy SOP等で使うことができる。 対応するローカル変数は\$UPX, \$UPY, \$UPZ (第1インプット用)および\$UPX2, \$UPY2, \$UPZ2 (第2インプット用)。
Keep / Add / No Particle Scale /pscale	各点のスケール属性の維持・追加・削除。 パーティクルのスケール設定は、Particle SOPでレンダリング属性として持たせることができ、最終的にはPoint Sprite MATでのスプライトのサイズなどで使われ

#### ■パラメータForce Page

Keep / Add / No Radius /radiusf	ロールオフ・エフェクトの距離に使用される。 ロールオフとは: $r / (r+d^2)$ rは半径、dはアトラクターからの距離。半径が設定されなければ減衰は行われない。
Keep / Add / No Force Scale /scalef	このアトラクターに関係する全フォースへの乗数(マルチプライヤー)。 半径とフォーススケールの両方で、デフォルトは1。
Radial / Normal / Edge / Directional Force	これら4つのパラメータは、作成されたときに一種のフォースを導入する。 各々、対応する乗数に関係している。
Radial Force /radialf	アトラクターへ向かうフォース。プラス値なマルチプライヤーならそちらへ向かい、マイナス値なら離れていく。
Normal Force /normalf	ポイント法線方向へのフォース。 Note: ポイントはポイント法線属性を元々は持っていない。しかし現状、計算された法線がオーバーライドされる。
Edge Force /edgef	Primitiveフェース・タイプでのみ作用するフォース。 フォースは、ポイントから導かれるエッジの向きへ方向づけられる。 多数の頂点が同じポイントを参照していた場合、方向はポイントを参照する最後のプリミティブのエッジ方向。 フェースが開いている場合、終点はそのプリミティブ中のひとつ前のポイントと等しいエッジ方向になる。 Note:ポイントSOPを使用して、エッジフォースが加えられる場合、フォース方向はポイントSOP自体の中で計算される。したがって、このSOPに続くどんな変形もこれらには影響しない。変形を加えたければ、すべてポイントSOPの前に行う必要がある。エッジフォースだけがこのように機能する。
Directional Force /dirf	距離ロールオフ機能によって影響を受けた上での、任意方向の力。

#### ■ローカル変数

\$PT	各点のポイント番号
\$NPT	トータルのポイント数
\$CEX, \$CEY, \$CEZ	SOPの中心位置。
\$TX, \$TY, \$TZ	各点の位置
\$BBX, \$BBY, \$BBZ	全ポイントを囲むバウンディングボックス内の相対位置。0から1で表される。
\$NX, \$NY, \$NZ	各点の法線
\$MAPU, \$MAPV, \$MAPW	各点に対応するテクスチャマップのUV値
\$CR, \$CG, \$CB	各点の色(R,G,B)
\$CA	各点のアルファ値
\$UPX, \$UPY, \$UPZ	上方向を示すベクトル(第1インプット用)
\$UPX2, \$UPY2, \$UPZ2	上方向を示すベクトル(第2インプット用)
以下はパーティクルに特化した変数	
\$PSCALE	各パーティクルのスケール
\$AGE	各パーティクルの秒齢
\$MASS	各パーティクルのmass
\$DRAG	各パーティクルのdrag coefficient
\$LIFE	各パーティクルの寿命中の齢(パーセント)
\$DIST	各点からintersected surfaceまでの距離。この属性はRay SOPによって作られる。
\$VX, \$VY, \$VZ	各パーティクルのVelocity
\$ID	各点のパーティクルID。この変数は、ポイントがパーティクルシステムに随伴されているときのみ定義される。

## Particle SOP

パーティクルを作ります。パーティクルシステムは雨や雪などの自然現象、花火などのシミュレートによく使われます。Touchでは、入力されたジオメトリのもつポイントをパーティクル発生源として使用する。それぞれのパーティクルは、外力(重力)や風の影響を受けます。またCollisionソースとして定義した物体や制限面にぶつかって跳ねさせる／消すこともできます。パーティクルは他の物体にはない様々な属性を持っています。velocity(移動ベクトル)、寿命・年齢などです。これらが各パーティクルに対して計算され、シミュレートを行うのです。

## ■パラメータState Page

Particle Type /prtype	パーティクルをどのように描画するかを選択
Render as Lines /lines	velocityに基づく長さを持つ2点の線として描画。
Render as Point Sprites /point	Point Sprite MATと共に使う。各パーティクルは常にカメラ方向を向いた四角形の絵として描画される。サイズは、PointSpriteと点のpscale属性によって決定される。PointSpriteテクスチャは絵の左下を原点に貼りつけられる。
Behaviour /behave	
Particle System /psystem	パーティクルシステムとして振る舞う。入力元SOPが持つポイントが発生源となる
Modify Source Geometry /modify	入力元の物体を変形する。 Tip: ジオメトリをインスタンスにすることもできます。SOPtoCHOPを使って、パーティクル位置をCHOP化し、GeometryCOMPのInstanceページでジオメトリを各パーティクル位置に置くことができます。これが多数のコピーを動かす最も有効なパーティクルが生まれたときと死んだときに、どのくらいのメモリを再利用するかの設定。
Point Reuse /ptreuse	リセットタイムになった後、どのくらいのシミュレーションの秒数をとるか。例えば、ここに33を設定すれば(そしてタイムラインがリセットされれば)、フレーム1は33秒分の過去のシミュレーションを反映する。言い方を変えると、最初の32秒はバイパスされ、33秒目がフレーム1にシフトされるということ。シミュレーションに時間がかかるので、リセット時にいくらかの遅れが出る場合がある。
Preroll Time /timepreroll	このパラメータは、どのくらいの頻度でこのSOPを計算(cook)するかを決定する。デフォルトでは「1/\$FPS」、つまり毎フレーム計算するという事です。複雑なダイナミクスが含まれる場合、正確性のためにより頻繁な計算が必要になるかもしれない。その場合は「1/\$FPS」以下の値を設定する。逆に「1/\$FPS」より大きくはリセットパラメータの解釈
Time Inc /timeinc	パーティクルシステムをリセット。0がOFF(リセット)で0以外がON。
Reset Condition /resetcondition	パーティクルの誕生するピクセルを不規則にする。
Reset /reset	フレームとフレームの間のパーティクルの軌道をも計算することによってより正確にパーティクルを動かす。
Jitter Births /jitter	入力元SOPが使われてないポイントを排除。シミュレーションの間、メモリからポイントをパージする時間を節約する。シミュレーションの間の不必要な減速を防ぐ。
Accurate Moves /accurate	
Remove Unused Points /rmunused	
Attractor Use /attractmode	
All Points	全てのアトラクター(引力を持つ)ポイントが全てのパーティクルに影響する。
Single point per particle	これが有効のとき、各パーティクルは1つのアトラクターポイントのみに割り当てられ、その影響のみを受ける。割当は、パーティクルとアトラクターポイントの同番号が対応(どちらかが多い場合は剰余で)。

## ■パラメータForces Page

External Force /externalx /externaly /externalz	重力としてパーティクルに作用。dragパラメータが0なら、パーティクルの加速には制限がない。
Wind /windx /windy /windz	風としてパーティクルに作用。External Forceと似ているが、風自体の速度を超えて加速はしない。 WindとExternal Forceの使い分けについて: External Forceの適用は、直接的にパーティクルの加速に影響し、その率はMass(F = Mass加速度)によって決まります。 Windはそれに加えての力ですが、velocityに敏感です。あるパーティクルが既に風力で運ばれているなら、それはそこからそれ以上の力を受けるべきではありません。風を使ったとき、これが最大のvelocityとなります。Mass(重さ)の増加は、加えられた力による加速を妨げます。Drag(抵抗)は加えられた力に相対する方向への力です。パーティクルの速度制限に便利です。)
Turbulence /turbx /turby /turbz	かき回し(混沌・無秩序)の強さを各座標軸について設定。正数値を設定すること。
Turb Period /period	小さくすると、かき回しは小さな単位で素早く行われ、それぞれの点がよりバラける。大きい値にすると、かき回しがゆっくり行われ、似た影響を受けるために、点どうしが近づくことになる。
Seed /seed	パーティクルシミュレーション用のランダムシード(乱数の基数)

## ■パラメータParticles Page

Add Particle ID /doit	各パーティクルに、誕生時にID番号を割り振る。
Add Mass Attribute /domass	選択すると、パーティクルのMass(重さ)を計算。
Mass /mass	各パーティクルの相対的なMass(重さ)。重いパーティクルは加速減速しにくい。
Add Drag Attribute /dodrag	選択すると、パーティクルのDrag(抵抗)を計算。
Drag /drag	各パーティクルのDrag(抵抗)。
Birth /birth	1秒に生まれるパーティクル数。1フレーム中に複数のパーティクルが生まれる場合、それらの出生時間は、ランダムなサブフレーム時間にセットされる。

Life Expect /life	各パーティクルの「寿命」。どのくらい存在するかの秒数。デフォルトは1000秒。これは16分半に相当し、大抵のアニメーションではこれより短くていいと考えられる。1とか2を入れて、効果を確かめてみるとよい。
Life Variance /lifevar	パーティクルの「寿命」のばらつきを秒数で指定。これを0にすると同時に生まれたパーティクルは完全に同時に消滅するが、例えば1を入力すれば、パーティクルによって消滅するタイミングに前後1秒のばらつきが生まれる。
Alpha Speed /alpha	パーティクルの速度が速くなるにつれて、パーティクルの透明度を上げていくことができる。デフォルトは0で、全く透明にならない。0.5にすると、1距離単位/秒の速度で70%の透明度になり、速度が増すにつれて透明になっていく。

#### ■パラメータLimits Page

+ / Limit Plane	LimitPlane(制限面)の定義。制限面にパーティクルが到達したときに、パーティクルは消滅するか跳ねるかする。前後左右上下で6面の制限面を定義できる。デフォルトでは1000単位も離れた場所にされている。
/limitposx ...	
/limitposz	
/limitnegx ...	
/limitnegz	
Hit Behaviour /hit	パーティクルが制限面またはコリジョン物体に衝突したときの振る舞いをコントロールする。選択肢は以下:
Die on Contact	衝突によって消滅
Bounce on Contact	衝突によって跳ね返る
Stick on Contact	衝突位置で停止
Gain Tangent / Normal /gaintan /gainnorm	衝突時の摩擦力によるエネルギー損失を表す。1つめのパラメータは、面に垂直の方向に維持されるエネルギー量で、0だと全てのエネルギー(速度)は失われ、1だと全く失われない(全反射)。2つめのパラメータは、接線方向のエネルギー量。例えば、これらを0.1と1にすると、屋根に落ちる雨粒のような動きに、1と0だと
Split /splittype	
No Splitting	パーティクルはそのまま(デフォルト)
Split on Contact	衝突によってパーティクルが分かれる。
Split on Death	パーティクルは消滅と同時に分かれる。
Min/max splits /splitmin /splitmax	パーティクルが分かれるとき何個に分かれるか。最小最大を決めればその間のランダムな数に分かれる。
Split Velocity /splitvelx /splitvely /splitvelz	別れた各パーティクルに与えられるベース速度。
Velocity Variance /splitvarx /splitvary /splitvarz	別れた各パーティクルに加えらるランダム速度。花火を作る場合はベース速度を大きくランダム速度は小さく、雨粒がはじけるのを作る場合はY方向へのベース速度とXZのランダム速度を大きくするとよい。

#### Texture SOP                      テクスチャマッピング設定

Texture SOPはUVおよびW座標を、テクスチャマッピングおよびバンプマッピングのためにソースジオメトリに割り付けます。これはテクスチャーのマルチレイヤー座標系を生成します。

#### ■Fixing Seams & Unrolling Geometry 継ぎ目の固定&ジオメトリの拡張

注:以下は、面およびhullプリミティブオブジェクトにのみ関係します。

テクスチャータイプが継ぎ目の固定を要し、テクスチャが頂点に適用される場合、ラップされたプリミティブはテクスチャー座標を計算する前に拡張されます。ラップされたプリミティブを広げることで、新しい頂点がそれら個別の頂点と同じポイントを使用する、開かれたプリミティブに変えます。したがって、拡張はポイント数を変更しません。また、割れ目をさらに現わさせません。

明示的な拡大(Primitive SOPの使用)は必要ありません。

継ぎ目固定はテクスチャー座標を計算した後に行われます。

テクスチャが頂点またはポイントに適用されても、それは必要とされる。また、それはu、vあるいは両方で行われます。

次の生地タイプが、継ぎ目の固定を要求します:

円筒状 - uに固定された継ぎ目

極 - uとvに固定された継ぎ目

列/峠 - 継ぎ目はuとvに固定

スプライン・タイプが「Uniform」「Chord Length」「Average」- uとvに固定された継ぎ目

注:プロジェクション方法が円筒状(Cylindrical)か極(Polar)の場合、閉じたメッシュ、Bzier & NURBS表面が開かれます。

頂点の少なくとも1つの列/カラムは加えられるでしょう。(たぶんNURBSより多く)

これは、接合個所の継ぎ目におけるテクスチャー座標の貧弱な描画を防ぎます。

#### ■パラメータ

Group /group	inputにグループがあった場合、グループ名を設定すればグループに属する要素のみに適用される。パターンマッチング(正規表現)記述が可能。
Texture Layer /texlayer	ジオメトリがマルチなテクスチャーレイヤを持っている場合、このパラメータでどのレイヤーにこのTextureSOPを影響させるか設定。
Texture Type /type	テクスチャ適用方法

	<p>スプラインベースのメソッドを選んだ場合、グループ欄の貼り付けの優先順位を明記すると、それらのノード全てへのテクスチャーコーディネートの計算に伝播する。Projectionメソッドは、どのスプラインメソッドよりもサーフェース間の継ぎ目がスムーズなテクスチャーを実現する。時には、貼り付けがBasis SOPとともにChord-lengthパラメータ化されるのを確実にするのに役立つ。XYZ軸からのダイレクトな投影(projection)。</p>
Orthographic	XYZ軸からのダイレクトな投影(projection)。
Polar	球体的にラップした投影
Cylindrical	円柱的にラップした投影
Rows and Columns	メッシュ構造のジオメトリ(Grid, Sphere, Tube, Skin, and Sweep)に対し、U座標はrowに、V座標はcolumnに沿う。これは、自動車のフェンダーのような、一方からのプロジェクションでは対応できない湾曲したメッシュに良い。
Face	全ての面にテクスチャーのコピーをマップ。これを使う前に、FacetSOPを使ってPointをuniqueにすべきです。マップは各面の法線に沿って投影される。しかし、このマッピングは、スケーリングを伴わず、また各ポリゴンの形状によって歪められる。ジオメトリのサイズが変わった場合、テクスチャーはジオメトリに「固定」されない。石垣やブロック壁など、ポリゴンとテクスチャーのエッジが合っていないものを投
Modify Source	ソースがすでにテクスチャーUVマップを持っていた場合、それは維持される。オフセットやスケーリングができる。 (補足)「UVマップ」というものの自体の説明: 3D物体の各点(X,Y,Z)と、絵(テクスチャー)上の各点(X,Y)との対応を「定義する」もの。絵の側の位置を「X,Y」と言ってしまうと混乱するので「U,V」座標と呼ぶ。3D-CGデータの各ポイント毎に、XYZ座標と共に(別に)、UV座標を持たせることができる。objなど主要な3Dデータファイルではそのデータを保持できるので、他ソフトに持って行っても、マップ画像を指定するだけで、貼り付け方は保たれる。なお、UV座標マップの設定はジオメトリに属する情報の定義であり、そこに実際に画像の貼り付けを行う工程とは通常は切り離されている。それが、TouchではTexture SOPとMaterial SOPということである。
Uniform Spline	NURBSとベジェ・サーフェースにのみ使用可能。テクスチャーが表面の点/頂点に調和するようにUとVで各々の表面の領域スペース(Basis)を試してみて、それらの(u,v)値を割り当てる。隣接した表面のテクスチャー・スペースの間で、連続性を確実にするためUやVでスプライン・ベースを合併するよう、Texture SOPの前にBasis SOPを挿入すること。
Average Spline	テクスチャー・アトリビュートに連続した結び目の平均を保持させる。これらの平均は、グレイヴィル点として知られている。この方法とUniform Splineは、ペーストされたサーフェースに推奨。
Arc Length Spline	スプラインの原則によったテクスチャーコーディネイトである点でUniform Spline方法と類似。両方の方法は同じ範囲でテクスチャー・コーディネイトを生み出し、最小・最大の結び目の値で制限する。違いは、連続したテクスチャー・コーディネイトの間に間隔にある。Average Splineは、一様にパラメータ・スペースを試してみる。Arc-Length Splineは、表面の弧-長に基づくテクスチャー・コーディネイトを選
Perspective From Camera	カメラ視点からのプロジェクションによるテクスチャーのコーディネイトが、正確にオブジェクト空間に投影される。どこかのポイントがnear clipping planeの手前だったり、far clipping planeより遠かったりした場合は、(0, 0, 0)のテクスチャーがアサインされる。

---

## Spring SOP

Spring SOPはポリゴンのエッジや各ポイントに付加された質量を元に、スプリング(ばね)の力(フォース)を用いて、入力されたジオメトリを変形・移動する。ジオメトリは、ポイントやエッジで簡単な物理シミュレートされた力によって変形されます。

シミュレートされた「質量(mass)」は、各ポイントに割り当てられます。

そのプリミティブのエッジは、フォースに対し、それらの元の状態の方へポイントを引き戻す「スプリング」として働きます。

スプリングがフォースによって伸ばされる場合、それらはポイントを引き戻そうとする。

ポイントはそれらが元の状態に戻るまで、それらの質量のためにびよよ〜んと揺れ続けます。

ポイントに作用する力は以下のとおり:

- ・外力(重力)
- ・風力(外力に似ている)
- ・タービュランス(無秩序な力)

より大きな抗力値、あるいはより小さな質量の場合、より早く発振は納まります。

---

## Force SOP

Force SOPは入力されたメタボールフィールドにforceアトリビュートを加えます。それは、Particle SOPかSpring SOPで、アトラクターやrepulsion force fieldsに使われます。一般的に、フォース値0以上は引力、0以下は斥力です。

### ■パラメータ

Radial Force /doradial	チェックされると、メタボール中央からの引力または斥力をトリガーする。
Force /radial	Radial Forceがチェックされているとき、これで強さを調節する。
Directional Force /doaxis	チェックされると、方向性のあるフォースに関する以下のパラメータを有効にす

Direction /dirx /diry /dirz	Directional Forceがチェックされているとき、これで方向を設定する。
Axial Force /axial	Directional Forceがチェックされているとき、プライマリな方向のフォースの強さを設定する。controls the force along the primary axis. Increasing this value will cause the particles to move up the primary axis of the metaball field as defined by the direction vector.
Vortex Force /vortex	Directional Forceがチェックされているとき、this field controls the amount of twist particles are given around the primary axis. Positive values cause the particles to spin clockwise, negative values cause counter-clockwise spins. It is a centrifugal force.
Spiral Force /spiral	Controls the attraction/repulsion force perpendicular to the primary axis (Direction field). Values greater than 0 will cause the points to be drawn toward the primary axis. Values less than 0 push particles away perpendicular to the primary axis. It is a tangential force.

#### ■入力

このSOPは、メタボールのジオメトリしか入力することができません。他を入力しても無効です。

#### ■Tips

メタボールフィールドの影響を視るには、入力されたメタボールのhull表示をONにする。Particle SOP と Spring SOPの両方が、hullをガイドジオメトリとして表示するでしょう。影響するメタボール・フィールド内のフィールド・エフェクトを増加/減少させるためには、メタボールのウェイトを調節してください。一般に、フォースとフォース・フィールドのスケールを維持しようと努めることはよい考えです。パーティクルの移動と吸引力を、より扱いやすく、よりリアルにできます。





3Dオブジェクトも、最終的には平面に投影されて初めてビジュアルになるのですから、映像ソフトにとって最も基本となるのが、二次元画像/映像をコントロールする部分と言えます。様々な合成・エフェクトが行えます。別にCGだからと言って必ずしも3Dを使わなくてもよく、むしろ2Dでいかに映像を操るかが映像のクオリティを決定します。

分類	OP名	説明	重要度	
新規生成系	Constant TOP	単色カラーマット作成	★	
	Text TOP	テキスト作成	★	
	Ramp TOP	グラデーション作成	★★	
	Noise TOP	ノイズ生成	★	
合成系	Composite TOP	各種合成(パラメータで合成方法選択可能)	★★	
	Add TOP	加算合成		
	Difference TOP	差分合成		
	Multiply TOP	乗算合成		
	Over TOP	普通に上に重ねる合成(アルファはちゃんと抜ける)	★	
	Under TOP	普通に下に重ねる合成(アルファはちゃんと抜ける)		
	Screen TOP	スクリーン合成		
	Subtract TOP	差分合成		
	Inside TOP	Input2の内側にInput1を配置する		
	Outside TOP	Input2の外側にInput1を配置する		
	Matte TOP	トラックマット合成。前景・背景に加えてマット画像を使って抜き方を指定できる。	★	
	Cross TOP	クロスフェード(ディゾルブ)	☆☆	
	Chroma Key TOP	クロマキー合成のためのキーイング	☆	
	RGB Key TOP	RGBカラー指定でキーイング		
エフェクト	Blur TOP	ブラー・ぼかし	☆☆	
	Luma Blur TOP	輝度または2番目の入力のグレースケール値に基づいてイメージにブラー		
	Corner Pin TOP	コーナーピン変形	☆☆	
	Edge TOP	エッジ(色の境目)を検出		
	Emboss TOP	エンボス。レリーフのように見せる。		
	Flip TOP	左右/上下反転		
	Monochrome TOP	モノクロ化		
	HSV Adjust TOP	画像をHSV(色相・彩度・輝度)方式で調整する		
	Level TOP	輝度・コントラスト・ガンマ調整	★	
	Luma Level TOP	輝度・コントラスト・ガンマ調整 with よりHueを維持するよう計算		
	Convolve TOP	周囲ピクセルとの重みづけ(「マトリックス」フィルタをDATテーブルで)		
	Tile TOP	画面をタイル状に増殖して並べる		
	Math TOP	計算		
	Displace TOP	ディスプレイースメント(置き換え)		
	Slope TOP	値と近隣するピクセルの値の差分を表すピクセルを生成		
	Lookup TOP	Input1の画像の色をInput2のカラーテーブルで置き換え		
	Feedback TOP	前フレームをフィードバックすることによって、モーションブラー効果を与える。(フィードバックしたものをそのまま重ねると、たちまち積み重なって真っ白になるのでLevelTOPで減衰させる、Transformで移動させるなどする。)	★★★	
	加工・変換	Crop TOP	クロップ。上下左右の不要部分の切り取り。	★
		Resolution TOP	解像度の変更	★
		Transform TOP	移動・回転・拡大縮小	
Channel Mix TOP		RGBAチャンネル加工		
Threshold TOP		ある閾値以上・以下の部分を抽出する	★	
RGB to HSV TOP		Rの値→Hue, Gの値→Saturation, Bの値→ValueとしてのHSVの画像に変換してしまう。チャンネルごとに加工する系のエフェクトで「Hueだけ加工」などといったことができる		
HSV to RGB TOP		上記のようにH,S,Vをチャンネル別に加工した後で元の色に戻すのに使う		
CHOP to TOP		CHOPチャンネル値を画像にする。1chの場合RGBAがその値に。4chの場合はそれぞれが	☆☆☆	
Reorder TOP		RGBAチャンネルを入れ替え		
Trim TOP		イメージシーケンスから1イメージを取り出す(あまり役に立たない)		
解析		Analyze TOP	画像ピクセルの平均・最大・最小を1xNのイメージで出力	☆☆
	Blob Track TOP	輪郭追跡(blob tracking)を行う。イメージの背景となる映像を最初に1分ほど学習させてから開始し「前面ピクセル」を見つけ出す。結果は、Info DAT/CHOPをアタッチすることで取り出す。	☆☆☆	
レンダリング	Render TOP	レンダリング。対象Geometry, Camera, Lightを指定して、3Dオブジェクトを「絵」として描画する。	★★★	
	Render Select TOP	RenderTOPからカラーバッファを取得		
	Render Pass TOP	マルチパス・レンダリング(1回で全てをレンダリングするのではなく、要素ごとにレンダリングして合成したりすること)を実現する	☆☆	

流れ制御	<b>Switch TOP</b>	入力映像のスイッチング。完全切り替えだけでなく比率での合成も可能。	★★★
	<b>In TOP</b>	コンポーネント外からの入力	★
	<b>Out TOP</b>	コンポーネント外への出力	★
	<b>Select TOP</b>	別のTOP(上階層でも下階層でもない)からの取り込み	
	<b>Null TOP</b>	ヌル(何もしない)	★
その他	<b>CUDA TOP</b>	NVIDIA GPUのCUDA DLLを記述	
	<b>Cache Select TOP</b>	Cache TOPにキャッシュされたイメージの取り出し	☆
	<b>Cache TOP</b>	映像をキャッシュ(一時記憶)する。短時間の録画を行うようなイメージ。	
	<b>Cube Map TOP</b>	Phong MATのリフレクションマップ用にキューブ・マップ(上下四方のマップ画像)を定義	
	<b>Depth TOP</b>	RenderTOPからデプスマップを取得	
	<b>Normal Map TOP</b>	画像から法線マップを生成。PhongMATとともに使う	☆
	<b>GLSL Multi TOP</b>	GLSL言語の記述による画像生成	
	<b>GLSL TOP</b>	GLSL言語の記述による画像生成	
	<b>SSAO TOP</b>	Screen Space Ambient Occlusion。デプスマップが必要なためRenderTOPと共に使われる	
	<b>OP Viewer TOP</b>	OP(Touchデザイン)画面そのものを取得	
	<b>Screen Grab TOP</b>	メインスクリーンのアウトプットを取得	
	<b>Sequence TOP</b>	複数インプットのTOPをイメージシーケンスにする？	
	<b>Time Machine TOP</b>	「モーフィング」が空間的にイメージをゆがめるのに対して、時間軸的にイメージをゆがめる	☆
<b>Texture 3D TOP</b>	CacheTOPに似ていてイメージシーケンスを記憶する。TimeMachine TOPと共に使う。	☆	
入出力	<b>Movie In TOP</b>	映像/画像ファイル入力	★★★
	<b>Movie Out TOP</b>	映像/画像ファイル出力	
	<b>Video In TOP</b>	USB/IEEEからの映像入力	★★★
	<b>Shared Mem In TOP</b>	他のTouchDesignerプロセスや他のアプリケーションとの共有メモリ・ブロックから画像データを	
	<b>Shared Mem Out TOP</b>	他のTouchDesignerプロセスや他のアプリケーションで使用のために共有メモリ・ブロックに画像データを出力	
	<b>Touch In TOP</b>	別Touchマシン(同一マシンも可)からのIn。TCP/IPで通信。Firewallに注意	
	<b>Touch Out TOP</b>	別TouchマシンへのOut	

## ●ほとんどのTOPで共通のパラメータ (Common Page)

Resolution	解像度(横×高さの画素数)	
- Input	入力解像度	
- Eighth, Quarter, Half, 2X, 4X, 8X	1/8~x8の解像度	
Custom Resolution	カスタム解像度	
Use Global Resolution Multiplier	メニュー>Edit>Preferences>TOPsで設定するグローバル解像度設定を使用。 パワーの違う別環境に持っていくときに有用。	
Aspect Ratio	画面アスペクト比	
- Input	入力に従う	
- Resolution	1:1ピクセルアスペクトとし、解像度設定に従う	
- Custom Aspect Ratio	カスタムアスペクト比	
Fill Viewer	TOPイメージをViewerでどう表示するかを選択	
- Input	入力	
- Fill	Viewerのサイズにストレッチする	
- Fit Horizontal	Viewerサイズに横フィット	
- Fit Vertical	Viewerサイズに縦フィット	
- Fit Best	Viewerサイズで切られないように縦横フィット	
- Fit Worst	Viewerサイズ一杯が画像になるように縦横フィット	
- Native Resolution	ネイティブ解像度を維持	
Viewer Smoothness	Viewのスムーズ化	
- Nearest Pixel	ニアレストピクセル法	
- Interpolate Pixels	平均補間	
- Mipmap Pixels	mipmapfiltering法。artifactsやsparklingを減らす。	
Pixel Format	色深度フォーマット指定 (ie. R, G, B, and A)	
- Input	入力に従う	
- 8-bit fixed (RGBA) ~32-bit float (A)		



\*エクスポート(他OPへのchannel値の出力)を行っているCHOPは「Exportフラグ」がONになります。

CHOPは、動きのデータ、音、画面上でのコントロール値、MIDI等外部機器からの入力など、常に変化していく値をコントロールできます。

ある1つの値(とその動き)を保持するものを「channel」といいます。

CHOP自体が値を持つわけではなく、CHOPオペレータの中をchannelが流れていく、と考えて下さい。

Channelの値を加工したり、流れを制御したりするのが(文字通り)オペレータの役割です。

そうやって作られた(変化する)値を、どんなノードのどのようなパラメータにも適用することができます。

(これをTouchDesignerでは「channelをExportする」と言います。)

例えば、3D物体の動きや形、大きさや色、映像へのエフェクトのかけ具合、鳴らす音の音量などが、channel値に応じて変化するような映像を作ることができるわけです。

また、そういったシームレスに変化するものだけでなく、値の条件によって表示するシーンそのものをカチッと切り替えるスイッチングや、スクリプト(プログラム)を実行開始したりもできます。

分類	OP名	説明	重要度	
新規生成系	Constant CHOP	新規定数チャンネル生成	★	
	Noise CHOP	ランダムパターン生成	★★★	
	Wave CHOP	基本波パターン生成	★★	
	Clock CHOP	システム時計を取得	★★	
	Beat CHOP	一定周期で生成(一時TimingCHOPと改名されてたが元に戻った)	★★★	
	Pulse CHOP	パルス波生成	★	
	LFO CHOP	(低周波数の)基本波パターン生成	★	
	Object CHOP	2つの3Dオブジェクトを比較して、絶対/相対的な位置と方向を値として出力するチャンネルを作成		
チャンネル自体の管理 流れ制御	Trigger CHOP	入力チャンネルがトリガーのしきい値を越えると、attack/decay/sustain/releaseのエンベロープを作成	★★★	
	Switch CHOP	スイッチング。インプット1の値に応じてインプット2~のどれかを出力	★★★	
	Merge CHOP	複数CHOP(内のチャンネル)を合流。チャンネル自体は混ぜられず、入力チャンネル数があるままキープされる。同一名チャンネルがあった場合はどれかをキープするか、別名にすることができる	★	
	Blend CHOP	チャンネル値のブレンド。入力1の値で入力2と3のブレンド割合をコントロールするなど。	★	
	Cross CHOP	クロスフェード用のチャンネルブレンド。複数入力(2つ以上)のチャンネルを比率で混ぜる。		
	Composite CHOP	CHOPのチャンネルを別のCHOPチャンネルにブレンド。Base、Layer、Effectの3入力がある。		
	Select CHOP	入力CHOP内の必要なチャンネルのみ取り出し	★	
	Delete CHOP	入力CHOP内の不要なチャンネルを削除	★	
	Lookup CHOP	入力インデックス値に応じて、いろんなCHOPから値を取ってくる	★★	
	Fan CHOP	チャンネル値(例えばch=1,2,3)をチャンネル(ch1,ch2,ch3)に分岐	★★★	
	Count CHOP	トリガーor閾値の通過を検知して、カウントしていく	★★	
	Speed CHOP	入力値を、指定した間隔でどんどん加算していく。リセットや、最大値に達した後の動きも指定	★★	
	Hold CHOP	トリガー時(2つめのインプットが0から1になった瞬間)のインプット1値を保持	★★	
	Feedback CHOP	フィードバックで使用するチャンネル値を蓄積	☆☆	
	Rename CHOP	チャンネルの名前変更	★	
	Replace CHOP	高速にチャンネルを取替。マッチしたチャンネルがInput2にある場合、Input1のチャンネルの出力は、Input2のチャンネルに取替		
	Reorder CHOP	1番目の入力チャンネルを数字あるいはアルファベット・パターンで再整理		
	Override CHOP	複数のCHOPソースの入力から、最新の更新チャンネルを出力		
	Copy CHOP	チャンネルの複製	★	
	In CHOP	コンポーネント外からの入力	★	
	Out CHOP	コンポーネント外への出力	★	
	Fetch CHOP	SOPやTOPなど別種のOPのパラメータをチャンネルとして取得	★	
	チャンネル値の加工	Math CHOP	計算を加える	★★★
		Function CHOP	Math CHOPより複雑な、三角関数、対数関数、指数関数の様な数学関数を提供	
		Logic CHOP	論理演算	★
		Delay CHOP	一定時間遅らせて通過させる。ディレイ	★★★
		Lag CHOP	ディレイ+スムージング	★
Limit CHOP		最大値・最小値を指定し、それを超える入力値を制限	★★	
Filter CHOP		channelをスムーズあるいはシャープにするフィルター		
Expression CHOP		インプットチャンネルをエクスプレッションで加工。複数インプット可能。 ic(input番号,channel番号,index)で、入力されたチャンネル値を取得(indexは横軸(時間軸)のこと) ローカル変数:\$C(チャンネル) \$NC(チャンネル数) \$Iインデックス \$V値	★★	
Spring CHOP		値の変化に対してばねのように元に戻ろうとする	★★	
チャンネル時間軸制御		Cycle CHOP	サイクル化	★
	Stretch CHOP	時間軸ストレッチ		
	Time Slice CHOP	サンプルのタイムスライスを出力。スムーズな補間を生成することに使用される。これを通してRecord CHOPやGesture CHOPに送ると、より滑らかにプレイバックできる。		
	Trim CHOP	時間軸トリム		
	Warp CHOP	タイムリマップ。第2入力のカーブで時間レートをコントロールする。		
	Extend CHOP	時間軸延長("extend conditions"を設定するだけ。リピートしたり)		

	<b>Shift CHOP</b>	startとendを変更して時間軸シフト		
	<b>Trail CHOP</b>	チャンネルは、ある瞬間の現在値しか持たないものだが、これに入れると、最近の値変化がグラフ表示される。(波形表示)		
	<b>Join CHOP</b>	時間軸方向に連結。入力全てに同じ数のチャンネルがあると想定。		
	<b>Gesture CHOP</b>	第1の入力の短いセグメントを記録してこの部分をループ化		
	<b>Event CHOP</b>	トリガーを受けて、その時点からのCHOPのシーケンスを生成する		
	<b>Interpolate CHOP</b>	複数の入力をキーフレームとして処理して、キーフレーム間を補間		
音系	<b>Audio In CHOP</b>	オーディオ入力	★	
	<b>Audio Out CHOP</b>	オーディオ出力	★	
	<b>Audio Play CHOP</b>	トリガーに従ってオーディオファイルを再生(全てメモリにロードされる) 効果音用に良い。AudioOutにつながなくてもこれだけで音が出る。	★	
	<b>Audio Stream CHOP</b>	オーディオファイルをストリーム再生(AudioPlayと違って全てロードしない)。URLも指定可能。BGM用に良い。AudioOutにつながないと実際に音は出ない。	★	
	<b>Audio Movie CHOP</b>	指定のMovieIn TOPの音声をMovieInと同期再生	★	
	<b>Band EQ CHOP</b>	イコライザー(周波数帯域に基づいて信号を強めたり弱めたりする)		
	<b>Parametric EQ CHOP</b>	パラメトリック・イコライザー		
	<b>Pass Filter CHOP</b>	バンドパスフィルター(周波数帯域に基づいて信号を通したり通さなかったりする)		
	<b>Pitch CHOP</b>	ピッチを検出		
	<b>Oscillator CHOP</b>	("音"レベルの周波数の)基本波パターン生成	★	
	<b>OP to Audio CHOP</b>	SOPやTOPを元にオシロスコープ風の音信号を生成		
情報取得	<b>Perform CHOP</b>	fpsのような、現在のTouchDesignerプロセスの状態を表すチャンネルを出力		
	<b>Panel CHOP</b>	TouchDesignerの各パネルの情報		
	<b>Render Pick CHOP</b>	Render Pass TOPからのレンダリングを元に、ジオメトリからピックアップした詳細な3D位置情報を返す。		
	<b>Info CHOP</b>	各ノード(OP)の、(パラメータの形になっていない)付加的な情報の取得。ここでしか拾えない情報も多い。(例えばエラーコードなど)		
外部との 入出力	<b>MIDI In Map CHOP</b>	MIDIの入力	★★★	
	<b>MIDI Out CHOP</b>	MIDI OUT	★	
	<b>Joystick CHOP</b>	ジョイスティック入力		
	<b>Keyboard CHOP</b>	キーボード入力		
	<b>Tablet CHOP</b>	WacomタブレットのXとY、ペン・プレッシャー、傾き、各種ペン・ボタンの値を取得		
	<b>Mouse CHOP</b>	マウス入力		
	<b>Mouse Out CHOP</b>	マウス出力(ポインタの強制移動)		
	<b>OSC In CHOP</b>	Open Sound Controlという形式のサウンドコントロール信号規格(iPhoneやMAXとかにも使われている)を入力		
	<b>OSC Out CHOP</b>	Open Sound Controlという形式のサウンドコントロール信号規格を出力		
	<b>Serial CHOP</b>	シリアルポート入力。Serial In DAT使ったほうが便利		
	<b>File CHOP</b>	CHOPで使用するチャンネルとオーディオ・ファイルを読み。Webからも可能		
	<b>File Out CHOP</b>	.chanまたは.logファイルにCHOPチャンネル・データを出力		
	<b>Shared Mem In CHOP</b>	自作アプリ(C++)と共有メモリ介してやりとりIn		
	<b>Shared Mem Out CHOP</b>	自作アプリ(C++)と共有メモリ介してやりとりOut		
	<b>Pipe In CHOP</b>	TCP/IP入力		
	<b>Pipe Out CHOP</b>	TCP/IP出力		
	<b>Touch In CHOP</b>	別TouchマシンからのIn		
	<b>Touch Out CHOP</b>	別TouchマシンからのOut		
	変換	<b>DAT to CHOP</b>	DATテーブルから値をCHOPへ。	
		<b>SOP to CHOP</b>	SOPのジオメトリオブジェクトからチャンネルを作成	
<b>TOP to CHOP</b>		TOPイメージのピクセルをCHOPチャンネルに変換		
解析・ 計算	<b>Analyze CHOP</b>	入力値の平均・最大・最小・合計・ピーク値などを計算		
	<b>Spectrum CHOP</b>	周波数スペクトルを計算。フィルターに通されたオーディオ信号を得るために、スペクトルを操作して逆変換可能		
	<b>Angle CHOP</b>	角度・ラジアン計算		
	<b>Slope CHOP</b>	入力チャンネルの導関数を算出(微分する)		
	<b>Transform CHOP</b>	t[xyz],r[xyz],s[xyz],p[xyz]と変換フォーム・オーダーによって定義されるような変換フォーム計算を行う		
	<b>Envelope CHOP</b>	ウィンドウ(対象範囲)を使用して、入力された各サンプル付近の最大振幅を出力。音量調整などに使う。		
その他	<b>Inverse Kin CHOP</b>	ボーン・オブジェクトのインバース・キネマティクス・シミュレーション		
	<b>Inverse Curve CHOP</b>	カーブ・オブジェクトを使用しているボーンのインバース・キネマティクス・シミュレーションを計		
	<b>Handle CHOP</b>	Handle COMPを使用しているインバース・キネマティクス・ソリューションを駆動するエンジン		
	<b>Attribute CHOP</b>	入力CHOPのアトリビュートを追加、削除、更新		
	<b>Keyframe CHOP</b>	キーフレーム管理(キーフレームをエディットしたいときはKeyframeCOMPを使う)		
	<b>Record CHOP</b>	チャンネルをリアルタイムにレコーディング。音もこれで録る。		
	<b>LTC CHOP</b>	SMPTEタイムコード(LTC)信号(音)をデコード		
	<b>Resample CHOP</b>	リサンプル		
	<b>Sequencer CHOP</b>	シーケンス・アニメーション用(現在はアルファ版。不使用推奨)		
	<b>Shuffle CHOP</b>	1組のチャンネルにサンプルを再編成。SOP to CHOPやTOP to CHOPを1つの行列だけのチャンネルに変換するなど		
	<b>Hog CHOP</b>	CPUタイムを無駄に使う(遅いPCでの動作をチェックするため)		

Null CHOP

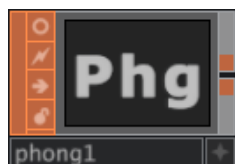
ヌル(何もしない)



## MAT

MATerial operator

## マテリアル(材質)を生成



マテリアルとは「材質」の意味で、3D物体に色やテクスチャマップを施す定義を行います。通常の立体表現ではほぼPhongMATの一択です。

分類	OP名	説明	重要度
シェーダー	<b>Phong MAT</b>	Phongシェーダー	★★★
	<b>Constant MAT</b>	単色カラーマテリアル	★★★
	<b>GLSL MAT</b>	C言語ライクな高レベルシェーディング言語GLSLによるカスタム・マテリアル(シェーダー)	☆☆
	<b>Wireframe MAT</b>	ポリゴンとカーブのエッジを線として描画	★★
スプライト	<b>Point Sprite MAT</b>	Point Spritesの(Particle SOPまたはDAT To SOPを使用して作成可能な)アトリビュートをコントロール	★★
その他	<b>Depth MAT</b>	Z方向に見えないオブジェクトを描画から除外するために使用される	

## Phong MAT

Phongシェーダー。テクスチャ、反射、バンプ、コーンライト、リムライト、アルファマップなどをサポートしている。Output Shaderパラメータを用いて、GLSL MATでより詳しい加工をすることもできる。Phongシェーディングでは3種のライトを使える。

Ambient 環境光  
 Diffuse 直接光。光源を向いた面が明るくなる度合い。  
 Specular スペキュラー。視点-物体-光源の関係で起きるハイライト。

## ■パラメータ

## RGBタブ

Ambient uses Diffuse /ambdiff	チェックONで、下のDiffuseとAmbientの設定値を同じにする
Diffuse /diff	Diffuseカラー
Ambient /amb	Ambientカラー
Specular /spec	Specularカラー
Emit /emit	光源が無くても表示されるカラー
Constant /constant	最終のカラーに加算されるカラー。最終の色 += 各ポイントの色 * Constant。Ambientとよく似た振舞いをするが、テクスチャや透明度の影響を受けない。
Shininess /shininess	スペキュラー量。増やすとツヤツヤ、減らすとマットな質感。
Color Map /colormap	カラーテクスチャマップ。TOPテクスチャを設定。このマップのアルファは、オブジェクトのアルファ計算に使われる。
Normal Map (Bump) /normalmap	バンプ(凹凸)マップ。法線を変化させることで疑似的な凹凸を作る。(使用方法)利用マップは、凹凸を明暗で表した画像をNormal Map TOPを通して処理したものを指定。対象SOPには、Attribute Create SOPで、tangent属性を付けること。
Bump Scale /bumpscale	バンプマップのスケール(強さ)
Diffuse Map /diffusemap	Diffuse量のマップ。このマップのアルファは無視される。
Specular Map /specmap	Specular量のマップ。このマップのアルファは無視される。
Environment Map /envmap	環境マップ。オブジェクトを囲む環境からの反射をシミュレートする。定義にはTOPテクスチャを用いる。環境マップは、普通のライティングに加算されるので、もし全反射させたければディフューズ・スペキュラは0にする。環境は球状マップされ、サンプルは以下。 <a href="http://www.pauldebevec.com/Probes/campus_probe.jpg">http://www.pauldebevec.com/Probes/campus_probe.jpg</a> TOPをCubeMapTOPにするか、RenderTOPの「RenderCubaMap」パラメータを使えばキューブ状のマップにすることもできる。
Polygon Front Face's /frontfacelit	ポリゴンの前面を照らすのに、ポリゴンの法線がどのように使われるかをコントロールする。詳細はTwo-Sided Lightingの記事を参照のこと。
Polygon Back Face's /backfacelit	ポリゴンの背面を照らすのに、ポリゴンの法線がどのように使われるかをコントロールする
Output Shader... /outputshader	このボタンは、このMATに影響するGLSL MATとそのコード記述用TextDATを作るダイアログを表示する。シェーダーはライトのタイプや数に依存するので、システム上で使われているライティング設定に基づくいくつかの可能なシェーダーを選びリストアップする。もしひとつもリストされなかったら、このセッションの中でレンダー可能なシェーダーがないということ。Phong MATのビューアーをONにするか、Render TOPでのレンダをセットアップすること。そうすればいくつかのシェーダーが作られ、リストに現れる。例えば、シャドウマップのシェーダーを見れば、シャドウマップを行うレンダーをセットアップすればリストに現れ

## ■パラメータ

## Alphaタブ

注: 単純に、アルファをオブジェクトに適用するだけでは、透明にはなりません。透過に関してはTransparencyの項を参照のこと。	
Alpha Map /alphamap	アルファマップ。画像のRedチャンネルが使われ、他は無視される。
Uniform Alpha /alphamode	これをoffにすると、アルファを、各ポリゴンの法線方向とカメラに依存して変える。 カメラの方を向いている法線について、下記Alpha Frontが適用され、カメラの横を向いている法線についてはAlpha Sideが適用される。
Alpha Front /alphafront	正面を向いたマテリアルの透明度。このパラメータとオブジェクトのPointアルファとが乗算される。(他のアルファソースも同様)
Alpha Side /alphaside	カメラに側面を向けたマテリアルの透明度。
Alpha Rolloff /rolloff	Alpha FrontからAlpha Sideへの変化のしかたをコントロールする。
Post-Mult Color by Alpha /postmul	計算の最後に、色(RGB)が計算されたアルファに乗算するか否かの設定。

Mult Alpha by Light Luminance /alcこれを有効にすると、ライティングによるルミノランスがアルファによって重ねられる。

### ■マルチテクスチャMulti-Texturingタブ

Multi-Texturingページでは、4つまでのテクスチャマップを保持して8パターンのテクスチャコーディネイトを選べる。デフォルトではそれらは乗算されるが、GLSLコードのフィールドも使える。ここは以下のように働く:

4つのテクスチャマップはそれぞれt0, t1, t2, t3というパラメータで参照される。なので、デフォルトの重ね方を表現すると「t0 \* t1 \* t2 \* t3」となるが、これを他の計算式にすることができる。例えば「t0 + (t1 \* 0.5)」といった具合。また「r.g.b.a」で各テクスチャのRGBAそれぞれを参照可能。例えば「t0 + (t1 \* (1.0 - t0.a))」。

Multi-Texturing (Disables Color Map) マルチテクスチャを可能にする  
 Texture [1-4] /texture1 through /t マルチテクスチャの4テクスチャを設定  
 Texture Coord /texture1coord through /tc どのテクスチャをマップとして使うかを設定する  
 GLSL Expression /multiexpr テクスチャイメージを統合するGLSLコード。空白のままでもよい。

### ■マルチテクスチャRimタブ

リムライトとは、撮影対象を挟んでカメラの反対側から強い光で照らすことで、輪郭を光らせて浮き上がらせるもの。縁明り。複数のリムライトを設定できる。(その場合、以下のパラメータ名の「1」が他のナンバーになる)

Enable Rim Light 1 /rim1enable リムライト有効設定  
 Rim Map /rim1map 計算されたリムライトの色に重ねるマップ  
 Rim Color /rim1color リムライトの色  
 Rim Center /rim1center リムライトの中心位置。360度の円周のどこか。  
 Rim Width /rim1width 中心からどのくらいリムライトが広がるか  
 Rim Strength /rim1strength リムライトの明るさ  
 Rim Strength Ramp /rim1strengthramp 水平方向のランプ(グラデーション)の強さをコントロール。

### ■パラメータ

#### Advancedタブ

Shadow Strength /shadowstrength This parameter will control how much being in a shadow will change the color of the lighting. At 1 the object will take on the Shadow Color parameter, at 0 it will behave as if it's not in a shadow, even if it is.  
 Shadow Color /shadowcolor 影の色  
 Darkness Emit darknessemit The Phong MAT calculates the current brightness of color of the objects, after taking into account lights, rim lights, emission etc. It then uses this brightness (between 0-1) and fades in the Darkness Emit Color. The darker the area, the more of the darkness emit color that will be applied.  
 Darkness Emit Color /darknessemit The color that is used for areas that are in darkness.  
 Darkness Emit Map /darknessemitt This map multiplies the Darkness Emit Color. This map's alpha is not used.  
 Darkness Emit Map Coord /darknessemittcoord The texture coordinate used to sample the darkness emit map.  
 Write Camera Space Depth To Alpha /writecamspacedeptoalpha This causes the camera space depth of the pixel to be written to the alpha channel of the output TOP. This value can be useful for post-processing effects, but of course you will not have the result of all the alpha calculations if you turn this on (although they'll get used to multiply the output color, assuming Post-Mult Color by Alpha is enabled).

### ■パラメータ

#### Optimizeタブ

Apply Point Color /applypointcolor 通常、SOPから来たカラー属性 (Cd[4]) は、ライティング計算に使われるが、このチェックを外すことで、その影響をOFFにできる。  
 Cubemap Normalization /cubemapnorm An alternate way of normalizing vectors in the shaders will be used, on some older cards this is faster than the default way of normalizing vectors. This is mostly a legacy feature now, but you can try it and see if your rendering speed improves.  
 Cubemap Resolution /cubemapres Determines the resolution of the cubemap that is used to normalize the vectors, higher resolution will result in higher quality, but uses up more texture memory.  
 The below parameters control whether cubemaps are used to normalize the named vectors.  
 Cubemap Normalize Light Vectors /cubemaplightvecs  
 Cubemap Normalize Half-Angle Vectors /cubemaphalfangles  
 Cubemap Normalize Eye Vector /cubemapeyevecs  
 Cubemap Normalize Spot Vectors /cubemapspotvecs  
 Cubemap Normalize Normal /cubemapnormal  
 The below parameters avoid doing extra work in the pixel shader for some vectors, resulting in faster rendering. However this can result in lighting artifacts. Try them out and see if they make a visual difference to your scene.  
 Fast Normalize Half-Angle /fastnormhalfang  
 Don't Re-normalize Normal /dontrenormnorm  
 Don't Re-normalize Light Vector /dontrenormlightvec  
 Don't Re-normalize Spot Vector /dontrenamespotvec  
 Don't Re-normalize Eye Vector /dontrenameeyevec  
 Fast Deform Tangent /fastdeformtang

### ■パラメータ

#### Deformタブ

詳しくはDeformの記事を参照。

Do Deform /dodeform このマテリアルでDeformを可能にする  
 Get Bone Data /deformdata Deformボーンデータがどこで獲得できるか



SOP with capture data /target sop	Specifies the SOP that contains the deform capture attributes/
pCaptPath Attrib /pcapt path	Specifies the name of the pCaptPath attribute to use. When your geometry has been put through a Bone Group SOP, the attributes will be split into names like pCaptPath0, pCaptPath1. You can only render 1 bone group at a time, so this should match the group you are rendering with this material.
pCaptData Attrib /pcapt data	Much like pCaptPath Attrib.
Skeleton Root Path /skelroot path	Specifies the path to the COMP where the root of the skeleton is located.
MAT /mat	When obtaining deform data from a MAT or a Deform In MAT, this is where that MAT is

## ■パラメータ

Commonタブ	
Blending	現在のカラーバッファのピクセルの色を加算。Blendingは透明のシミュレートによく使われる。 blendingの式: 最終的なピクセル値 = (ソースBlend * ソースカラー) + (目的Blend * 目的カラー)
Blending (Transparency) /blending	BlendingのON/OFF
Source Color /srcblend	This value is multiplied by the color value of the pixel that is being written to the Color-Buffer (also known as the Source Color).
Destination Color /destblend	This value is multiplied by the color value of the pixel currently in the Color-Buffer (also known as the Destination Color).
Depth Test	Depth-Testing is comparing the depth value of the pixel being drawn with the pixel currently in the Frame-Buffer. A pixel that is determined to be in-front of the pixel currently in the Frame-Buffer will be drawn over it. Pixels that are determined to be behind the pixel currently in the Frame-Buffer will not be drawn. Depth-Testing allows geometry in a 3D scene to occlude geometry behind it, and be occluded by geometry in-front of it regardless of the order the geometry was drawn. For a more detailed description of Depth-Testing, refer to the Depth-Test article.
Depth Test /depthtest	Depth-TestのON/OFF. If the depth-test is disabled, depths values aren't written to the Depth-Buffer.
Depth Test Function /depthfunc	The depth value of the pixel being drawn is compared using to the depth value currently in the depth-buffer using this function. If the test passes than the pixel is drawn to the Frame-Buffer. If the test fails the pixel is discarded and no changes are made to the
Write Depth Values /depthwriting	If Write Depth Values is on, pixels that pass the depth-test will write their depth value to the Depth-Buffer. If this isn't on then no changes will be made to the Depth-Buffer, regardless of if the pixels drawn pass or fail the depth-test.
Alpha Test	Alpha-testing allows you to choose to draw or not draw a pixel based on it's alpha value.
Discard Pixels Based On Alpha /alpha	Pixel alpha testのON/OFF
Alpha Threshold /alphathreshold	This value is what the pixel's alpha is compared to to determine if the pixel should be drawn. Pixels with alpha greater than the Alpha Threshold will be drawn. Pixels with alpha less than or equal to the Alpha Threshold will not be drawn.
Wire Frame	The wire-frame feature will render the geometry as wire-frame, using the actual primitive type used in the render. What this means is surfaces like Metaballs, NURBs and Beziers will become a wire-frame of the triangles/triangle-strips used to render them (since these types of primitives can't be natively rendered in OpenGL).
Wire Frame /wireframe	ワイヤースクリーンのON/OFF
Line Width /linewidth	ラインの太さ。数値はピクセル数。
Cull Face	The cull face parameter will cull faces from the render output. This can be used as an optimization or sometimes to remove artifacts. See Back-Face Culling for more
Cull Face /cullface	Selects which faces to render.
Use Render Setting	use the render settings found in the Render or Render Pass TOP.
Neither	do not cull any faces, render everything.
Back Faces	cull back faces, render front faces.
Front Faces	cull front faces, render back faces.
Both Faces	cull both faces, render nothing.
Polygon Depth Offset	This feature pushes the polygons back into space a tiny fraction. This is useful when you are rendering two polygons directly on-top of each other and are experiencing Z-Fighting. Refer to Polygon Depth Offset for more information. This is also an important
Polygon Depth Offset /polygonoffset	polygon offset機能をONにする





TouchDesignerのノードによるネットワークは、階層化(パッケージ化)が可能です。  
(起動して最初に現れるネットワークも、コンポーネントの内部です。)  
また3Dオブジェクト自体はSOPで定義しますが、レンダリングするには、Geometryコンポーネント内にまとめたものが対象となります。  
パネル部品など、コンポーネントの形で提供されている「部品」がたくさんあります。

分類	OP名	説明	重要度
基本形	Container COMP	コンテナ。ネットワークを内包できる。	★★★
	Base COMP	パネル関係や3D関係の機能が無いコンテナ。データ系ネットワークをまとめるのに使うとよ	
3D空間 関係	Geometry COMP	ジオメトリ	★★★
	Camera COMP	カメラ	★★★
	Camera Blend COMP		
	Light COMP	ライト	★★★
	Ambient Light COMP	アンビエント(環境)ライト	
	Blend COMP		
	Xform In COMP	自作アプリ(C++)と共有メモリ介して3DやりとりIn	
	Xform Out COMP	自作アプリ(C++)と共有メモリ介して3DやりとりOut	
	Bone COMP	ボーン	
	Handle COMP	IKハンドル	
GUI (パネル)	Button COMP	ボタン定義	★
	Slider COMP	スライダー定義	★
	Field COMP	テキストフィールド定義	★
	Select COMP	セレクター定義	
	Window COMP	フローティングウインドウ定義&マルチモニターへの映像の最終出力に用いる。	★★★
その他	Animation COMP	キーフレームアニメーション用のCHOP+DATのコンポーネント	
	Keyframe COMP	キーフレーム定義システム	★
	Table COMP	TableDATに様々な機能を内包して使いやすくした部品	
	Replicator COMP		
	Time COMP	タイムライン管理	
	Shared Mem In COMP	自作アプリ(C++)と共有メモリ介して値やりとりIn	
	Shared Mem Out COMP	自作アプリ(C++)と共有メモリ介して値やりとりOut	
	Null COMP	ヌル(何もしない)	★

#### ■Cloneについて

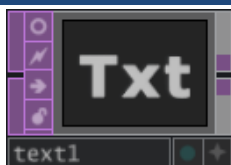
Cloneは、マスターとなるコンポーネントの構造や振る舞いを様々なコンポーネントにマッチさせるTouchのメカニズムです。  
マスターコンポーネントを変更すれば、クローンに反映されます。  
コンポーネントの「Clone」パラメータは、マスターへのパスです。もしこの項が空白なら、それはクローンではないということです。

・クローンがそのマスターと共通するもの：  
全てのクローンは、同じOPを内包します。  
内部OPのネットワークの繋がり方、パラメータ、フラグ状態も同じになります。

・各クローンがUniqueに保持できるもの：  
コンポーネントそのもののパラメータは、それがクローンであるかどうかに関わらずそれぞれが持ちます。  
各クローンのcookと入出力はそれぞれで行われます。入出力が異なるということは内部で処理されるデータも変わるということです。  
またクローン内の変数では、\$OPN (operator parent's name) や \$OPD (digits in operator's parent's name)が各クローン毎に異なります。  
これらの変数を用いて処理が変わるロジックを組み込めば、クローンごとで異なる処理を走らせることができます。

#### ・immune(免疫)ノード

クローン・コンポーネントの内側にあるノードでありながら、マスター下の相当OPからは独立性を保つノード。ノードでimmuneフラグが立っているものは、クローンのマスターの変更反映を受けません。例えば、コンポーネント内のTableDATをimmuneにすれば、各クローンコンポーネントで別々の値を保持できます。  
ノードがimmuneノード化された後は、マスター下でパラメータを変えても、マスター下で削除したとしても、影響を受けなくなります。



テキストデータかテーブル(表)形式のデータを保持・加工します。  
 テキストにはスクリプト(プログラム)を記述できます。  
 アルゴリズムが複雑化した場合は、全てをネットワーク構築で考えなくても、プログラムしたほうが早いです。  
 テーブルに関しては、プログラム経験者ならば「配列」がいかに便利かを御存じでしょう。  
 様々な値を配列管理でき、またDB的な検索・ソートもできます。

分類	OP名	説明	形式	重要度
基本系	Table DAT	テーブル(セル)式DAT	Table	★
	Text DAT	テキスト式DAT	Text	★
加工系	Indices DAT	テーブルで一連の番号を作成	Table	
	Merge DAT	テキストまたはテーブルをマージ	Table	
	Sort DAT	行か列にテーブルDATデータをソート	Table	
	Reorder DAT	入力テーブルの行と列を並べ替え	Table	
	Transpose DAT	入力DATテーブルの列を行に変換	Table	
	Substitute DAT	入力DATテーブルの全セルに対する文字列置換	Table	
	Evaluate DAT	セルに計算式を書きこんだテーブルを第2入力に入れて、第1入力のテーブル上の値を計算する	Table	☆
	Insert DAT	テーブルに列またはカラムを挿入。入力DATがテーブルでない場合はテーブルに変換	Table	
	実行	Execute DAT	入力に接続されたTextDATのスクリプトを実行	Text
CHOP Execute DAT		CHOPの値変化によって、入力に接続されたTextDATのスクリプトを実行	Text	★
DAT Execute DAT		他のDATテーブルの値変化によって、入力に接続されたTextDATのスクリプトを実行	Text	
OP Execute DAT		OPの変化によって、入力に接続されたTextDATのスクリプトを実行	Text	
Panel Execute DAT		パネルコンポーネントによって、入力に接続されたTextDATのスクリプトを実行	Text	☆
Parm Execute DAT		あらゆるノードのパラメータの状態が変更された時、入力に接続されたTextDATのスクリプトを実行	Text	
その他	Clip DAT			
	FIFO DAT	先入れ先出し方式でデータを保持		
	Info DAT	いくつかのノード(OP)が持っている付加的な文字列情報を取得。Blob Track TOPなどは、分析結果をこれを通す形で拾うようになっている。		
	Perform DAT	DATテーブル形式でパフォーマンス・タイムを記録		
	Render Pick DAT			
変換	CHOP to DAT	CHOPチャンネル値をテーブル・フォーマットでDATに取得	Table	
	SOP to DAT	ポイント、プリミティブデータ、アトリビュートを抽出	Table	
	Convert DAT	テキストDATとテーブルDATの相互変換	-	
入出力	Serial In DAT	コンピュータのUSBまたはRS-232ポートでシリアル通信くアートナイトでCO2センサーに使用した	Table	
	TCP/IP In DAT	2台のコンピュータ間のTCP/IP接続について情報を送受信	Table	
	UDP In DAT	2台のコンピュータ間のUDP接続について情報を送受信	Table	
	Web DAT	HTMLデータを取得	Text	
	XML DAT	XML、SGML/HTMLフォーマットデータを解析	Table	
	OSC In DAT		Table	
	MIDI In DAT	MIDIインプット	Table	
	MIDI Event DAT		Table	
	Multi Touch in DAT	Windows7標準のマルチ・タッチAPIからメッセージとイベントを受取	Table	
	File In DAT	ファイル入力	-	
	File Out DAT	ファイル出力	-	
フロー制御	Select DAT	TouchDesigner内のあらゆる場所からDATを参照	-	
	Switch DAT	入力パラメータを使用して、出力を選択するマルチ入力オペレータ	-	
	In DAT	コンポーネント外からの入力	-	
	Out DAT	コンポーネント外への出力	-	
	Null DAT	ヌル(何もしない)	-	

#### ■DAT Export

各OPのパラメータをテーブルで管理できる。やり方は簡単で、TableDAT上に2列作って1列目にOP名/パラメータ名、2列目に値を書き込み、エクスポートフラグ(右下●)をONにするだけ。

#### ■スクリプト記述

スクリプトをテキストDATに記述し、トリガー実行するようにすれば、何でもできる。

(例)ButtonCOMPを作り、その中でAudio Play CHOPを作る。次にExecute DATを作ってViewerをActiveにする。

Viewerをクリックして"audioplay audioplay1 0"と記入。

DATの"Execute"メニューパラメータを"On Panel Change"にする。uで上位階層へ行きボタンのViewerをActiveにしてクリックすれば音が鳴る。なおスクリプトのrun命令で他のExecute DATのスクリプト実行もできる。

#### ■テーブルの使い方

Select DAT, Evaluate DAT, Merge DAT, Switch DAT, Sort DATなどで加工ができる。

またスクリプトのtabcellコマンドで加工でき、セルはtab(), tabr(), tabc(), tabrc()などのエクスペッションで読める。

エクスペッションの書き方はHelp -> Commands and Expressions参照。

(例)Table DATを作ってViewerをActiveにする。右クリックAdd Column, Add Rowで2列3行にする。

name	age
joe	9
jane	21

と打ち込む。次にTextTOPを作り、Text/パラメータに、`tabc("table1", \$OD, "name")`と入力する。TextTOPには"joe"と出ているでしょう。

このTextTOPをCopy/pasteすると、新しいTOPには"jane"と出ます。このtabc()はテーブルから値を取得するエクスペッションで、

カラム名は1行目のもの、「\$OD」は"Operator Digit"を意味し、オペレータ名についての数字を表す。

## TScript

## スクリプト

## Variables

分類	OP名	説明	重要度
グローバル	\$F	カレントフレーム	★★★
	\$T	カレントタイム = (\$F-1)/\$FPS	
	\$HOME	この.toeファイルのあるパス(ファイルパス記述用)	★★
Copy SOP	\$CY	コピーごとの番号	★★
	\$NCY	コピー総数	★
Point SOP (Also Copy SOP as Template)	\$PT	Point番号	★
	\$NPT	Point総数	
	\$TX, \$TY, \$TZ	Pointの位置	
	\$NX, \$NY, \$NZ	Pointの法線	
	\$ID	パーティクルID	★★
	\$PSCALE	パーティクルサイズ	★
	\$AGE	パーティクルの生誕からの経過秒数	★
	\$VX, \$VY, \$VZ	パーティクル方向	
	Expression CHOP	\$C	チャンネル番号
\$NC		チャンネル数	
\$I		チャンネルインデックス(時間軸)	
\$V		チャンネル値	

## Expressions

分類	tabr("OPパス", 行名, 列番号)	説明	重要度
パラメータ 取得	par("OPパス/パラメータ名")	あらゆるOPのパラメータ取得	★
	var("OPパス","変数名")	OPローカル変数を外から取得(\$不要)	
	chop("OPパス/チャンネル名")	CHOPチャンネル値取得	★
	chopf("OPパス/チャンネル名",frame)	任意フレームにおけるCHOPチャンネル値	
テーブル値 取得	tab("OPパス", 行番号, 列番号)	テーブルのセルに入れた値の取得。番号は0から。	★
	tabr("OPパス", 行名, 列番号)	行・列に名前前でアクセスしたい場合はこれらの関数	
	tabc("OPパス", 行番号, 列名)		
	tabrc("OPパス", 行名, 列名)		
	tabnr("OPパス")	テーブルの行数を取得	
	tabnc("OPパス")	テーブルの列数を取得	
条件式	if(条件,真のときの値,偽のときの値)	条件によって返す値を変える。どこのパラメータ欄にも直接書けるので、これで結構いろいろできる。	★★★
	ifs(条件,真のときの文字列,偽のときの文字列)	戻り値を文字列にしたいときのif()	
数値関数	max(値1,値2), min(値1,値2)	大きいほうの値を返す・小さいほうの値を返す	★
	rand(基値)	0~1の乱数	★★★
	int(float値)	小数点以下切り捨てによる整数化	★
	round(float値)	四捨五入による整数化	
	abs(値)	絶対値	
	sin(角度),cos(角度),tan(角度)	サイン・コサイン・タンジェント	
	sqrt(値)	平方根	
文字列関数	strcat(文字列1,文字列2)	文字列を連結	
	strcmp(文字列1,文字列2)	文字列比較。0なら一致。-1なら<, 1なら>。	
	substr(文字列,開始位置,文字数)	文字列から一部を取りだし	
	strlen(文字列)	文字列長を返す	
	expand("変数名入り文字列")	変数名の部分とその内容に展開した文字列を返す	
	digits(文字列)	数字付きの文字列の数字を取出 例:digits("chan123") = 123	
その他	noise(valX,valY,valZ)	ジオメトリ用のノイズ加え用関数	
	execute("コマンド文字列")	Commandを実行してその結果を返す。	
	smooth(\$F,スタートフレーム,エンドフレーム)	スタートフレームのとき0、エンドフレームのとき1になるようなスムーズなフロート数を返す	
	pulse(\$F,スタートフレーム,エンドフレーム)	スタートフレームとエンドフレームの間でパルスとなる 0>1>0を返す。全部\$Fにすれば実行の瞬間パルス。	

## Commands

分類	tabr("OPパス", 行名, 列番号)	説明	重要度
パラメータ 操作	opparm [-p] OP指定 パラメータ名 (value(s))	「パラメータ名 (value(s))」を複数列挙可能 (例)opparm geo1 tx (1) ty (2) tz (3) オプション[-p]:パルスとして送信。この値を一旦OPに入れてcookし、元の値に戻すことができる。 (例) opparm -p moviein1 reload (1) :moviein1をリロード	★★★

テーブル値 操作	tabcell OPパス 列指定方法 行 列 値	テーブルの(row,col)に値をセットする 列指定方法: [RC] 行名・列名 / [rc] 行番号・列番号 [Rc] 行名・列番号 / [rC] 行番号・列名 行名・列名にはワイルドカード使用可能(当てはまるところ全てにセット)	★
	table オプション OPパス	テーブル自体の操作 オプション: [-a r]行追加 [-a c]列追加 [-d] [-r row_name] [-c col_name]行や列の削除	
その他	set	変数セット	
	opadd	新しいOPを作る	
	opwire	OP間の連結	
	opset	OPのフラグをセット	
	run	他スクリプトを引数付きで実行	
	include	他スクリプトを取り込み	
	click	TouchDesigner control panelへの仮想クリック	
	find	検索。modes, parameters, scripts, variables, etc.	
	viewfile	view a file that TouchDesigner supports	
	cc	カレントcomponentの移動(cdみたいなもの)	
	lc	component内をリスト表示(lsみたいなもの)	
	system	Windowsコマンドの実行	
	controlpanel	(obsoleted by the Window component) bring up floating window with many display options.	
ヒストリ	!!	Repeat last command	
	!str	Repeat last command matching str	
	!num	Repeat command "num" from the history list	
	!-5	Repeat the command run five commands previous	

#### ■囲み記号について

シングルコーテーション '': 完全に単なる文字列とみなされ、展開(解釈)されない。

ダブルコーテーション "": 囲み内部に書かれたことが展開(解釈)される。OP名やch名。

バッククオート ` `: 囲み内部に書かれるのはTouch Expressionとみなされ、実行(計算・評価)され結果がここに入る。`expression(parameter)`  
ただし、書く場所がOP内のパラメータで、かつ単純な数値を返すものである場合は、省略可能。

(例)Geometryのtxパラメータに書くときにわざわざ`sin(\$F)`と書かなくていい。

#### ■変数

変数とExpressionの評価は同時に行われる。変数は「\$」に変数名を続けることで(変数だということを)認識される。

変数名は英数字およびアンダーバー( )。(先頭文字は英字のみ)

{ }で囲むことで、変数名を変数で表現できる。例えば、abc1,abc2,abc3...という変数を作り、変数iに数値が入った状態でabc\${i}と書くと、iの値に応じてabc1~のどれかにアクセス可能になる。配列代わりになる。

#### ■演算子

C準拠。論理演算子は== != < > <= >= - && ||。Notは^。剰余は%。

#### ■スクリプト内コメントは#

#### ■条件分岐

```
if (expr) [then]
else if (expr2) [then]
else
endif
```

#### ■ループ

```
for variable = start to end [step increment]
end
```

```
foreach variable (element_list)
end
```

```
while (expression)
end
```

#### ■テキストポート画面で、OS(シェル)のように直接コマンドを打ったりもできるし、

シェルスクリプトを作って実行したりもできる。UNIX的！

結果文のリダイレクト出力「> >>」も可能(TextDATへ出せばよい)