

AN INTRODUCTION TO TOUCHDESIGNER

NVOID ART-TECH LIMITED

Copyright © 2015

PUBLISHED BY NVOID ART-TECH LIMITED

hello@nvoid.com

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en_US.

First printing, June 2015

クレジット

菱形 千恵 - Chie Hishigata - 翻訳

Alelo Translation - info@alelotranslation.com

ベン・ヴォイト - Ben Voigt - 編修

Derivative - www.derivative.ca

特別な感謝の意を表して

松波 直秀 - Naohide Matsunami - 校正

antymark - www.antymark.com

グレッグ・ハマノビック - Greg Hermanovic - Derivative社 創立者

Derivative - www.derivative.ca

エルバース・ソーカビ - Elburz Sorkhabi - 著者

nvoid - www.nvoid.com

目次

0 序章	6
0.1 本書の発行に寄せて	6
0.2 まえがき	7
0.3 TouchDesignerとは	9
1 基本 操作	10
1.1 データの流れと繋がり	10
1.2 オペレーターの作成	11
1.3 マウスとキーボード	14
1.4 ネットワークとパス	15
1.5 外部のテキストエディタの使用	16
1.6 ヘルプ機能	17
2 ユーザーインターフェイス	19
2.1 パラメータウィンドウ	19
2.2 パラメータ	20
2.3 トランスポートコントロール	23
2.4 タイムラインセッティング	23
2.5 ペイン（作業領域）	24
2.6 Palette Browser（パレットブラウザー）	26
2.7 検索ダイアログボックス	28
2.8 リアルタイムモード	28
2.9 便利なキー操作	29
3 TOP	30
3.1 はじめに	30
3.2 Movie In TOP	30

3.3	ムービーファイルのプリロード（事前読み込み）	30
3.4	Null TOP と Select TOP	31
3.5	コーデック	32
4	CHOP 35	
4.1	はじめに	35
4.2	データの送受信	35
4.3	音声の入力・出力	36
4.4	サンプルレート	37
4.5	タイムスライシング	38
5	DAT 40	
5.1	はじめに	40
5.2	通信方法	40
6	SOPs 42	
6.1	はじめに	42
6.2	レンダリング	42
7	COMPs 47	
7.1	はじめに	47
7.2	Window COMP	47
7.3	ユーザーインターフェイスコンポーネント	48
8	MATs 50	
8.1	はじめに	50
8.2	Phong、GLSL、Point Sprite マテリアル	50
8.3	UVマップ	51

0 序章

0.1 本書の発行に寄せて

アーティストは「創造」という挑戦に立ち向かうために型にはまらない思考と良きツールを必要とする。そして、アーティストとそのツールを結びつけるのは計り知れない程のテクニックと幅広いノウハウであろう。

エルバーズがTouchDesignerについての本を書くこと耳にした時、「彼が知っている事を是非知りたい！」という思いが沸き起こった。

彼がTouchDesignerについての奥深い知識を持っていたのは明らかであったからだ。多くのユーザー達とフォーラムで会話をしては手助けをしていたし、時間に余裕がないプロジェクトを集中して短期間で仕上げる事も良くあった。そして、TouchDesignerのベータ版を使ってDerivative社と共同で仕事をしたこともあった。更に、彼はヒップホップトロンボーン奏者とビデオジョッキーとしての顔も持っているのだから、なぜ僕がそう感じたかは言うまでもないだろう。

この本はTouchDesignerの概念・操作方法・テクニックを追及する、パソコンもしくはベッド脇の供として最適だろうと思う。私には今までこれらをユーザー達に伝える機会がなかったけれど、こうしてエルバーズが本の執筆を提案してくれて嬉しい限りである。この本のおかげで、新しいテクニックを学んで腕に磨きをかけるのも更になるだろう。最初から最後まで順に読んでもいいし、ランダムにページを開きながら新発見をして行くのも面白いと思う（自分は気づけばこうしていた）。

TouchDesignerコミュニティーに強く貢献するであろうこの本の執筆を先導・実行してくれたエルバーズに感謝したい。読者からの素晴らしい反応が、更に素晴らしい効果を呼ぶだろうと期待している。

Derivative社

創作者

グレッグ・ハマノビック (Greg Hermanovic)

0.2 まえがき

この本を書くとした目的は大きく分けて二つ有ります。

- ・ TouchDesignerの基礎を説明すること
- ・ ビギナーのために、ユーザーコミュニティの意見が反映される資料を設けること

まず一つ目に、この本を通して、ユーザーインターフェイス、オペレーター、オペレーターファミリー、合理的なワークフロー、ネットワークの最適化、パフォーマンスインターフェイス、ディスプレイ管理等色々なトピックを説明します。

この本の終わりではサンプルプロジェクトや説明ビデオを使いながら様々な課題を実践で解決します。これらのレッスンではプロジェクトやモジュール等を実際に一から作成するのでとても実用的です。

二つ目の目的は、ユーザーコミュニティからの助けがあつてこそ成り立ちます。私は、TouchDesignerコミュニティが力を合わせて素晴らしい結果を築いてくれると信じています。だからこそ、サンプルファイルや説明ビデオを含むこの本の内容全ては無料で提供されていますし、それだけでなく、ユーザーの皆さんも編集・制作活動に参加することができます。

誰にでも、この本の内容の編集、もしくはサンプルファイル等の追加を行うことが出来るということです。この本の内容は全て、LaTeXで書かれていて、編修に使われるコードやプロジェクトファイル、原稿、図形、ビデオ等あらゆるコンテンツがGitHubに保存されています。クリエイティブ・コモンズ・ライセンスを使用していますので、無料でダウンロード、シェア、そして編集が可能なのです。

誰でもレポジトリにブランチを加えて編集内容を入力する事は出来ませんが、クオリティー保持の為、nVoidスタッフのレビューを通った変更のみがメインブランチに統合されます。編集に貢献して頂いた全ての方の名前は、感謝の意を表してこの本のクレジット（協力）欄に記載されます。こうすれば、編集を統合した一つの方稿を読むことで、全ての読者がコミュニティの最新の知識に触れることができる訳です。

コミュニティが一丸となって、この本を初心者ユーザーの為の究極のハウツー本に作り上げる事を願っております！

そして、この本ではTouchDesignerの様々なテクニックを紹介するものの、決してTouchDesignerのユーザーマニュアルではないということをご理解頂きたいと思います。一つ一つ全ての機能

の説明をする訳ではありませんし、Derivative社のWikiページやフォーラムの代わりになる物でもありません。選ばれた項目のみを取り上げて説明しています。

最後に、この本はTouchDesignerのプログラマー、そしてDerivative社のスタッフ達への贈り物だと思っています。何故なら今の私達ユーザーの知識は彼らのフォーラム上での、もしくは実際に会った上での手助けから成り立っているのだから。この素晴らしいコミュニティー一環の助け合いの精神がTouchDesignerが使われている限り続く事を願っています。

エルバース・ソーカビ (Elburz Sorkhabi) 及び nVoid Art-Tech Limited社

0.3 TouchDesignerとは

使い始めは、ほとんどのユーザーがTouchDesignerとは一体何なのか、まずこれを知りたいと思います。単純なタスクを作成・実行するのに長時間がかかることにも、他のソフトでは当たり前を用意されている機能を構築するのに結構な時間がかかることにも驚くかもしれない。

TouchDesignerとは一体何なのか？答えは至ってシンプルで、TouchDesignerとはノードベースのビジュアルプログラミング言語です。

まず始めに、最も重要な「TouchDesignerがプログラミング言語であること」、これを説明しましょう。TouchDesignerは他のソフトでは当たり前と感じられる機能を提供するものではありません。とてつもない可能性を秘めていて、そしてそれと同時に落とし穴がある可能性もある環境なのです。慣れと経験の積み重ねで、様々な機能の作成時間は短縮されるはずです。この本のゴールである再利用可能なモジュールの作成を目指している内に、真っ白で手を触れられていないキャンバスが完成プロジェクトになる時間はかなりのペースで短縮されるはずです。それでもTouchDesignerがプログラミング言語であることには変わりはないので、慎重に準備する時間と労力はもちろん必要だし、ユーザー達が頭を抱えることは良くあるはずです。ほとんどの場合、どれだけ柔軟に解決策を見つけ出せるかが鍵になるでしょう。

二つ目は、「TouchDesignerがノードベースであること」です。何十、何百行にも渡るコードをテキストで打ち込む代わりに、ノードベースのビジュアルインターフェイスを使ってアプリケーションを作成します。それぞれのノード（TouchDesigner内のオペレーター）が極微ながらも特定の作業を行います。難度の高いタスクを実行するには数々のノードが同時に作動します。データを入力・出力する為には、複数のノードがワイヤーで繋がれます。Cycling 74のMax/MSP等、沢山のノードベースのプログラミング言語がありますが、他と比べてTouchDesignerが勝っている点はビジュアルに関してだと言えるでしょう。

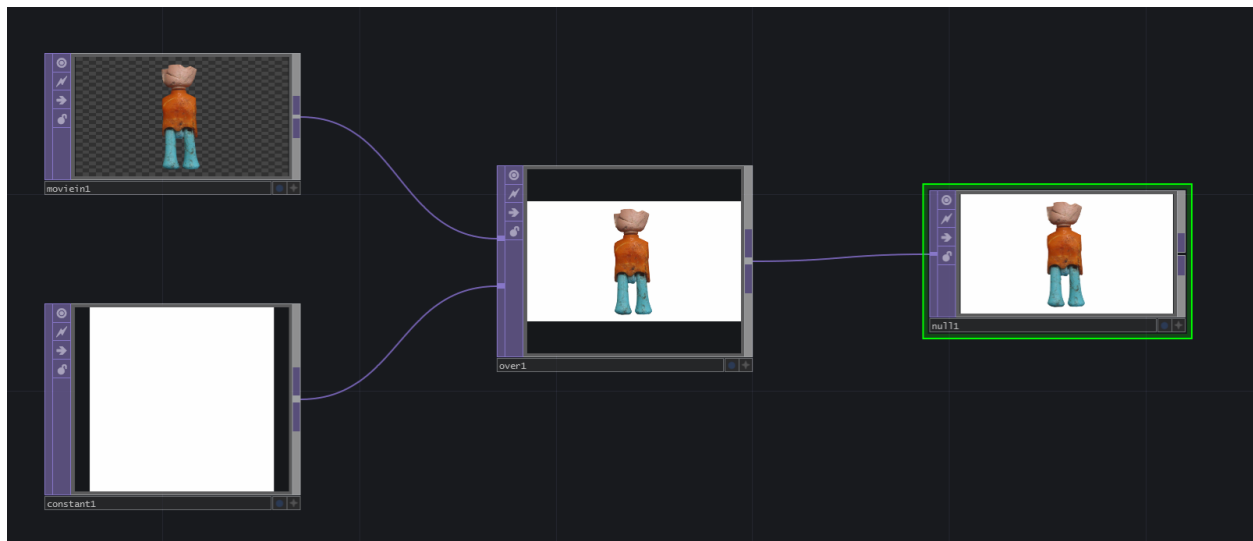
TouchDesignerを操作する時、全ては映像化されて表示されます。全てのオペレーターにはビューアーが備わっていて、テキスト、コントロールデータ、音声、ビデオ等、例え何であろうと一つ一つの実行された作業が映像によって表示されます。こういった点は従来のプログラミング、そしてノードベース言語とは異なりますが、それがTouchDesignerの素晴らしい特徴だと言えるでしょう。一つ一つのステップを映像で見ることによって、複雑なタスクも簡単に習得して実行することができることでしょう。

1 基本操作

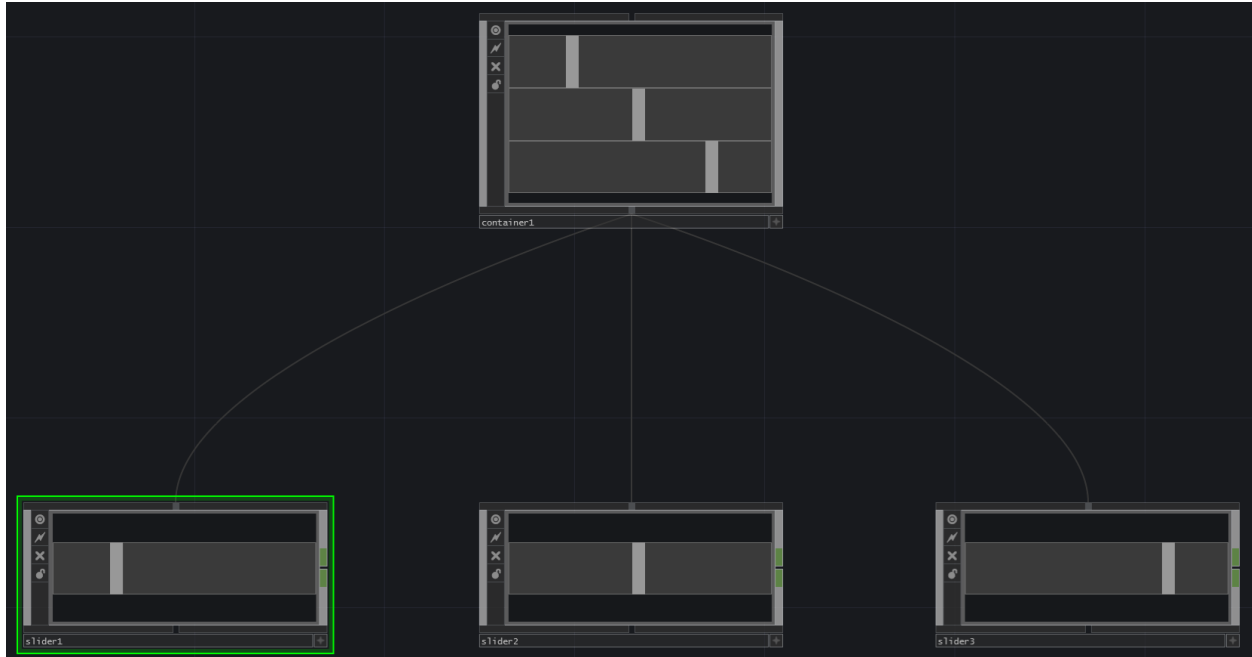
1.1 データの流れと繋がり

全てのプロジェクトはワイヤーで繋がれた数々Operator（オペレーター）で構成される為、オペレーターをワイヤーで繋ぐのがTouchDesignerで行う最も基本的な作業と言えるでしょう。それぞれのオペレーターが独自の機能を果たしますが、複数のNetwork（ネットワーク）として繋がれたその時、とてつもなく複雑な課題を達成・解決することできるのです。

TouchDesigner内のデータは全て左から右へと流れていきます。入力データは左側に、出力データはその右側へと順に並べられます。オペレーターの側面にある入力コネクタと出力コネクタは最初に繋がれた物から順に上から下へと並べられます。例えば、下のサンプル画面では二つのデータの流が左から右へ行くに伴って合成されていきます。



Component（コンポーネント）を使用する場合でもデータは同様に左から右へと流れていきます。オペレーターと異なる点といえば、コンポーネントでは親子関係を築くことが可能だということです。親子関係は上から下へと並べられます。一番上にあるコンポーネントが親、その下のコンポーネントは子、その更に下のコンポーネントはまたその子（孫）、という感じで繋がっていきます。下のサンプル画面では、三つのスライダーを表示しているContainer COMP（コンテナコンポーネント）のインターフェイスがあります。このContainer COMPが親で、その下のSlider COMPがその子供たちになります。



1.2 オペレーターの作成

いつ、どの画面で使用するかにもよりますが、OP Create ダイアログボックスを表示するにはいくつかの方法があります。オペレーターを一から作成する場合、最も簡単な二つの方法は1) キーボードの「Tab」キーを押すこと、そして2) ネットワークの背景（画面の何も無いところ）をダブルクリックすることです。

既に繋がっている複数のオペレーター（オペレーターチェーン）に新規オペレーターを追加する場合は次の三つの方法が有効です。一つ目はオペレーターの入力コネクタ又は出力コネクタの部分をクリックする方法です。この方法で、入力コネクタの前、もしくは出力コネクタの後ろに直接オペレーターを追加することが出来ます。既存のオペレーターチェーンに簡単に追加することが出来る便利な方法です。

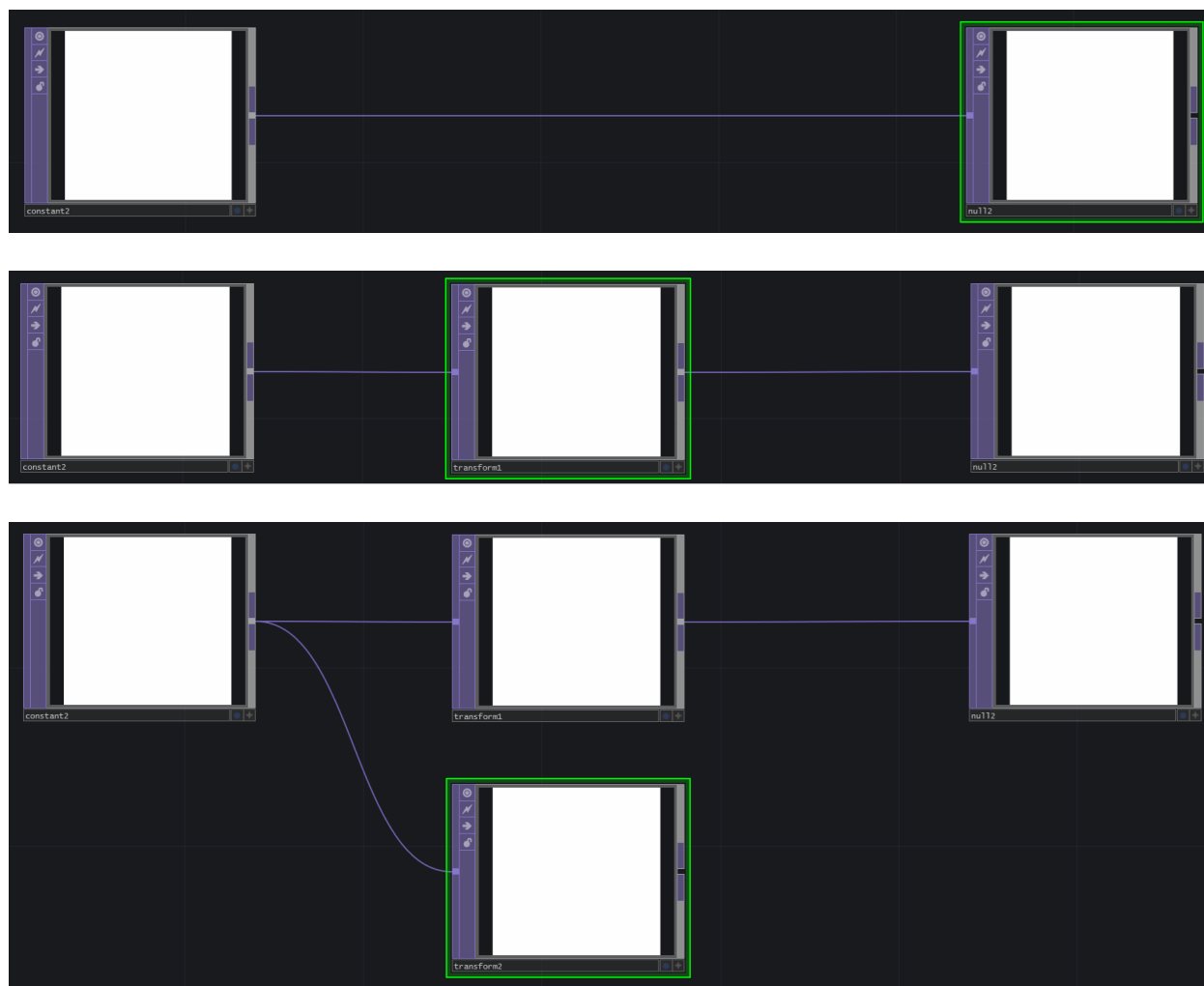
例えば、Constant TOPとNull TOPが繋がっていて、その間にTransform TOPを加えたいとしましょう。Constant TOPの出力コネクタ、またはNull TOPの入力コネクタをクリックしてTransform TOPを選択することによって、新規オペレーター（Transform TOP）を既存の二つの間に追加することが出来ます。

コネクタの代わりに既存の二つのオペレーターを繋ぐワイヤーをクリックして「Insert Operator」を選択することによって、同様に新規オペレーターを間に追加することもできます。

オペレーターを作成する三つ目の方法はワイヤーをクリックして「Add Operator」を選択する方法です。先程の「Insert Operator」との違いは、「Insert Operator」（挿入）は新しいオペ

レータを既存のオペレーターチェーン内に一体化させるのに対して、「Add Operator」（付加）では新たに枝分かれしたチェーンが作成され、新規オペレーターはその先に追加されます。

下のサンプル画面で説明しましょう。一番目の画面にはConstant TOPとNull TOPが一つずつ表示されています。次の画面では間のワイヤーを右クリックして、「Insert Operator」を選択することによってTransform TOPが挿入されました。そしてその次の画面では、ワイヤーの右クリック後「Add Operator」を選択することによってTransform TOPと新たなワイヤーが付加されています。二つの違いが分かるかと思います。



オペレーターを作成する時に使える、ControlキーとShiftキーの便利なキー操作をご紹介します。OP Create ダイアログボックスを開いてControlキーを押したまま、複数のオペレーターオプションを選択すると、ネットワークの下部に、ワイヤーで繋がれていない状態のオペレーターが一気に追加されます。この方法は複数のオペレーターを素早く追加するのに便利な方法です。

次は、OP Create ダイアログボックスでShiftキーを使ってみましょう。Shiftキーを押しながら複数のオペレーターオプションを選択した場合、Controlを使った場合とは違い、オペレーターが既にワイヤーで繋がった状態で追加されます。この方法では様々なサイズのオペレーターチェーンを素早く作り上げることが可能です。

ご紹介した二つのキー操作はそれぞれとても便利な物ですが、二つを組み合わせ使用した場合、更に素晴らしい成果が得られます。例を挙げて説明しましょう。三つのオペレーターチェーンを作成するとします。一つ目はCircle TOP、Blur TOP、Null TOPの順で繋がれたチェーン、二つ目はCircle TOP、Edge TOP、Noise TOP、Null TOPのチェーン、そして最後三つ目はNoise TOP、Blur TOP、Null TOPで構成されているチェーンだとします。

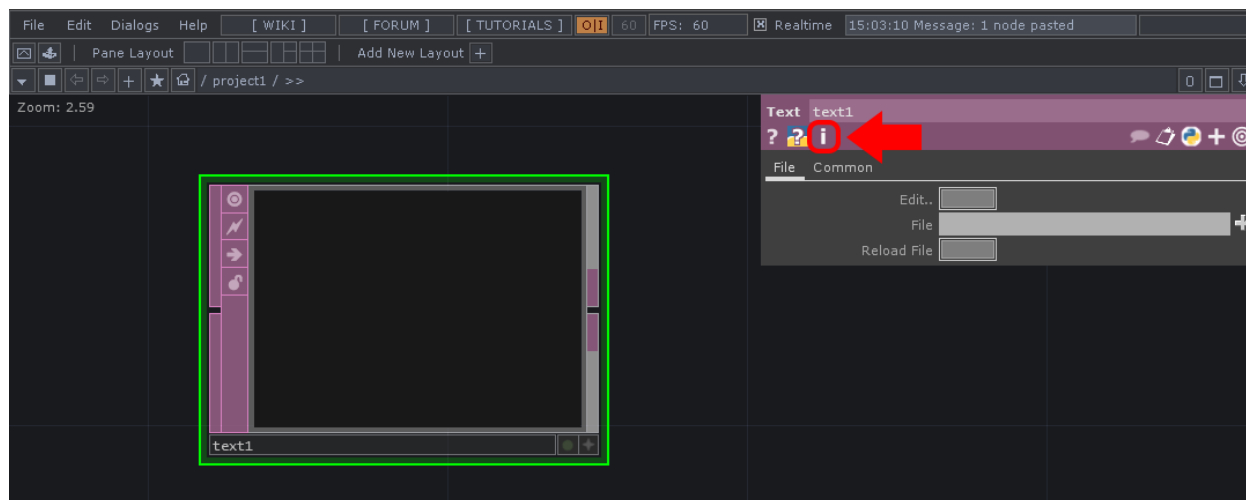
1. OP Create ダイアログボックスを開きます。
2. Shiftキーを押します。
3. Shiftキーを押したまま、Circle TOP、Blur TOP、Null TOPを選択します。これで一つ目のチェーンが作成されます。
4. Shiftキーから指を離します。
5. Controlキーを押します。Controlキーを使うと、ワイヤーで繋がれていない状態のオペレーターを追加することになります。
6. Controlキーを押し続けた状態で、Circle TOPを選択します。
7. Controlキーから指を離します。
8. Shiftキーを押します。
9. Shiftキーを押しながら、Edge TOP、Noise TOP、Null TOPを選択します。これで二つ目のチェーンが出来上がります。
10. Shiftキーから指を離します。
11. Controlキーを押します。
12. Controlキーを押しながら、Movie In TOPを選択します。
13. Controlキーから指を離します。
14. Shiftキーを押します。
15. Blur TOPと Null TOPを選択します。
16. 全てのオペレーターが作成されたので、Esc（エスケープ）キーを押してOP Create ダイアログボックスを閉じます。

三つの全てのオペレーターチェーンが素早く作成され、後はOP Create ダイアログボックスを閉じるだけでプロジェクトの仕上げに集中することが出来ます。二つのキー操作を使用することにより、オペレーターを作成する度にOP Create ダイアログボックスを開け閉めする必要がなくなる上に、個々のオペレーターを手動で繋ぐ時間も短縮できます。

1.3 マウスとキーボード

オペレーターを操作するにも、ネットワーク内を移動するにも、TouchDesignerではマウスはとても重要な役目を果たす為、高性能のマウスの使用をお勧めします。

ネットワークの背景を左クリックしてドラッグするとネットワークを動かしてスクリーンに対しての位置を調整することが出来ます。また、左クリックでオペレーターを選択し、左クリックしてドラッグすることによってオペレーターをネットワーク内で移動することが出来ます。オペレーターを右クリックすれば様々なメニューが表示されますが、メニューの詳細は後ほどゆっくりとご紹介します。複数のオペレーターを同時に選択したい場合は右クリックとドラッグすることによって点線選択エリア内の全てのオペレーターが一気に選択されます。オペレーターをマウスの中央ボタンでクリックするとそのオペレーターについての詳細情報が表示されます。中央ボタンがないマウスを使用する場合には、下のサンプル画面に表示されている「i」ボタンを左クリックすることによって同じ情報を表示することが出来ます。



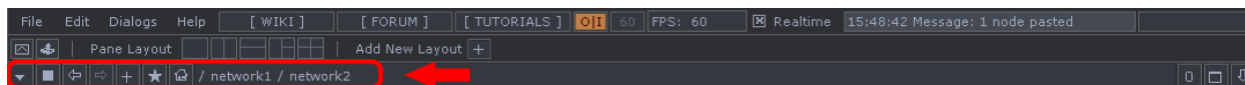
TouchDesigner内の移動に良く使われる二つのキー操作「u」と「i」をご紹介しましょう。「u」キー（up）を押すことによって現在選択中のコンポーネントから一レベル上のネットワークへと素早く移動できます。次に、「i」キー（in）を押すと選択中のネットワークまたはコンポーネント（Container COMPやBase COMP等）の中に移動することが出来ます。

全てのオペレーター（ネットワークの全体）をスクリーンに表示したい場合は、「h」キー（home）を押します。ネットワーク（何も表示されていないところ）の右クリックメニューで「Home All」を選択しても同じ結果が得られます。

1.4 ネットワークとパス

TouchDesignerのプロジェクトは全てネットワークによって構成されています。複数のオペレーターがワイヤーで繋がれてネットワークが出来上がりますが、無制限の数のオペレーターとワイヤーでとても複雑に構成されていることも良くあります。そして、ネットワークはContainer COMP、Base COMP、Geometry COMP等のコンポーネントの中に入れて整理することが出来ます。TouchDesignerの最上階レベルの画面は「root（ルート）」と呼ばれ、ここにはシステム用にデフォルトで作成される二つのオレンジ色のコンポーネントがあります。

プロジェクトを制作してから直ぐにでもネットワークをコンポーネントに入れて整理することをお勧めします。現在のパスはNetwork Editor（ネットワークエディター）の上部にあるパスバー（下の例で赤で囲まれている部分）に常に表示されています。



全てのオペレーターにはパスがあります。Unixファイルのパスに類似するものであり、

「Absolute path（絶対パス）」と「Relative path（相対パス）」の二種類があります。「絶対パス」はroot及び「/」からの現在使用中のオペレーターの位置づけを示すもののに対し、「相対パス」は現在使用中のオペレーターからもう一つ別のオペレーターまでの道筋を示すものです。「相対パス」の場合、パスのスタート地点はrootではなく、現在使用中のオペレーターが属するネットワークとなります。

具体的な例として、「Path.toe」ファイルを開いて見てください。TouchDesignerがrootレベルで「network1」と呼ばれるContainer COMPを表示した状態で開きます。Network1の中には二つのオペレーター、「rel1」と「Network2」があります。「rel1」はText DATで、中には二つのパスが表記されています。一つ目は「絶対パス」で、プロジェクトの最上階レベルであるrootから始まり、現在地のオペレーターに向かっての道筋を示します。二つ目は「相対パス」で、現在地「rel1」から目的地である「rel2」までの道筋が表記されています。「rel2」はText DATであり、Container COMPである「Network2」の中に位置します。「rel2」に到達するためには、「Network2」内への移動が必要となるため、「相対パス」は「network2/rel2」となります。

「Network2」のオペレータービューアーがネットワーク内にあるText DATのオペレータービューアーをそのまま表示しているのに気づいてもらえるかと思います。先にパスの説明を優先して、

このテクニックについては後ほど例をあげながら詳しく説明します。「Network2」の「Operator Viewer」パラメーターに「./display」と表記されています。オペレーターの名前が「display」で、そして「./」は現在使用中オペレーターの一段階内側のレベルを指しますので、この相対パスは現在のオペレーター「Network2」から「display」への道筋を示していることになります。

「network2」内の「display」という名のText DATのコンテンツが一レベル上のネットワークに表示されているのです。その他にも参照できるパスの例を挙げると、「abs1」にも絶対パスが表記されていますし、「rel2」には二つの相対パスの例が表記されています。一つ目は「abs1」への相対パスで、二つのオペレーターは既に同じネットワーク内に位置しているので「abs1」（オペレーターの名前のみ）が相対パスになります。二つ目の「../rel1」は「rel1」への相対パスで、「rel1」がオペレーターの名前で、「../」は現在地であるネットワークからもう一レベル上のネットワークを指しています。「../」は最上階レベルであるルートに到達するまで繰り返して使えるものの、他にももっと効率的な方法を後ほどご紹介します。

1.5 外部のテキストエディタの使用

TouchDesigner内で比較的短めのPythonスクリプトが作成・編集できるものの、スクリプトが長くなるに連れて外部のテキストエディタを使用した方が作成時間を大幅に短縮することができます。

数々の便利機能が外部のテキストエディタでは使える訳ですが、全部をここに書き出す代わりに、代表的なものをご紹介します。

1. 行番号表示
2. シンタックスハイライト
3. 検索置換機能
4. オートコンプリート機能

長いスクリプトを作成・編集する場合、これらの機能によって仕事の大幅な効率化に繋がります。

外部のテキストエディタをセットアップする手順は次の通りです。

1. 「Edit」メニューから「Preferences」を開きます。
2. 「DAT」タブを左クリックします。
3. 「Text Editor」オプションの隣にある参照ファイルのアイコンを選択します。
4. 使用したいテキストエディタの実行ファイル（.exeファイル）を選択し、「Open」を左クリックします。通常、実行ファイルは「Program Files」フォルダーに保存されています。

5. 「Accept」 ボタンを左クリックします。

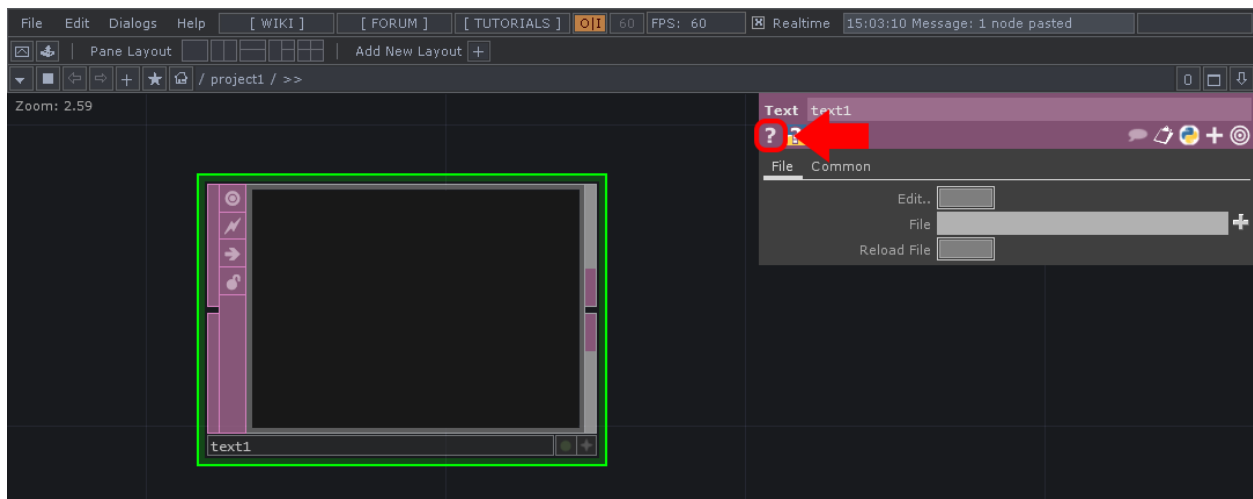
セットアップ完了後、試しにText DATを作成し、そのText DATを右クリック、そして「Edit Contents...」（コンテンツの編集）を選択してみてください。「Preferences」で選択したテキストエディタが開くはずです。Table DATの編集の為に外部エディタを使用する場合は、同様に「Preferences」で「Table Editor」を選択し、外部エディタを指定します。

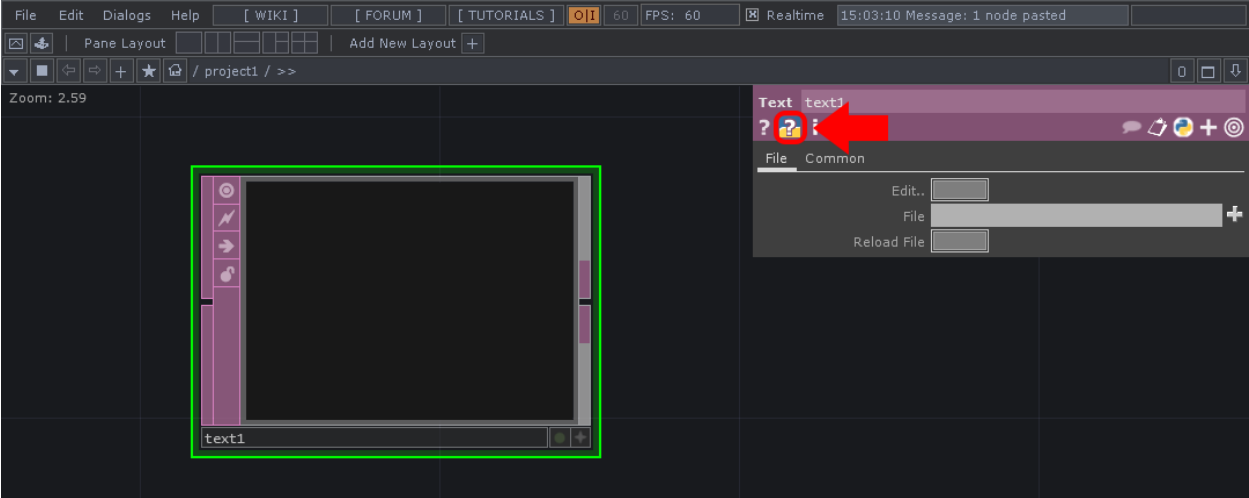
TouchDesignerユーザー達に頻繁に使用されていて評判の高いクロスプラットフォームエディタは次の二つです。

- 1) Sublime Text 3 <http://www.sublimetext.com/>
- 2) Notepad++ <http://notepad-plus-plus.org/>

1.6 ヘルプ機能

使用中オペレーターについて疑問がある場合はいつでもDerivative社のWikiページで詳しい説明を読むことができます。パラメータウィンドウ内の二つの？ボタンで現在使用中のオペレーターについてのWikiページを新規ブラウザウィンドウで開くことができます。一つ目（白の？）は使用中のオペレーターについて、二つ目（Pythonロゴと白の？）は使用中オペレーターのPythonスクリプトについての詳細情報を表示します。





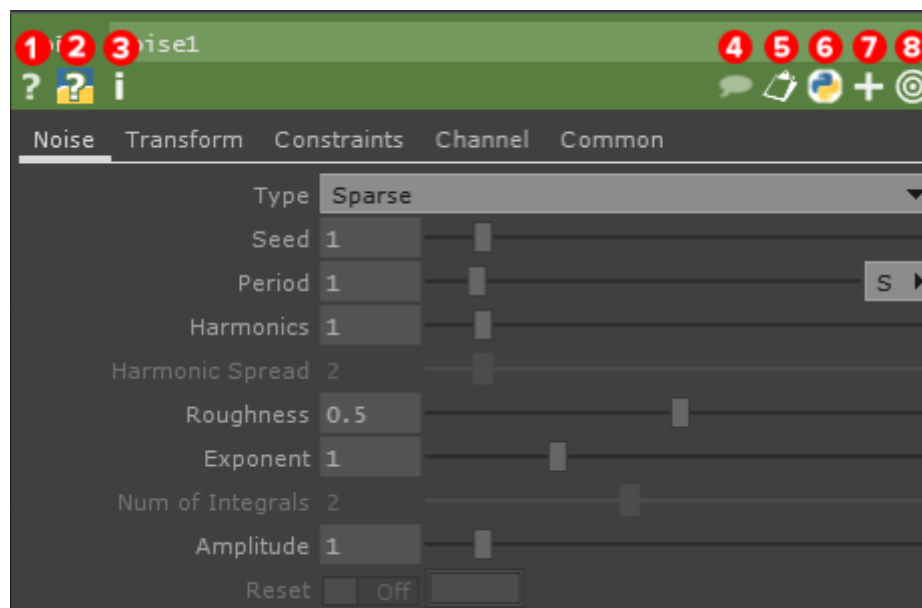
2 ユーザーインターフェイス

2.1 パラメータウィンドウ

オペレーターへ様々な調整を加えることが出来るパラメータは「パラメータウィンドウ」内に並べられています。パラメータウィンドウを開く方法は二通りあり、一つ目はキーボードの「p」キーを押す方法です。これによって選択中のオペレーターのパラメータウィンドウが画面右上に開きます。別のオペレーターを選択した場合には、表示内容は新たに選択されたオペレーターのパラメータへと自動的に変わります。

二つ目はオペレーターを右クリックして「Parameters...」を選択する方法です。この場合はパラメータウィンドウがフローティングウィンドウとして開きます。一つ目の方法とは違い、たとえば他のオペレーターを選択しても、終始初めに右クリックしたオペレーターに関するパラメータのみを表示し続けます。複数のオペレーターのパラメータウィンドウを同時に開きたい場合に便利な方法です。

調整可能なパラメータはオペレーターの種類によって変わりますが、オプションアイコンは全てのパラメータウィンドウにおいて共通のものが用意されています。こちらのサンプル画面でオプションアイコンをご紹介します。



左から右に向かって、次の通りです。

1. オペレーターヘルプ： そのオペレーターについてのWikiヘルプページがブラウザーで開きます。
2. オペレーターPythonヘルプ： そのオペレーターについてのPython Wikiヘルプページがブラウザーで開きます。
3. 詳細情報ダイアログボックス： オペレーターを中クリックした場合と同様に、そのオペレーターについての詳細情報が表示されます。
4. コメント： オペレーターについてのコメントを表示・編集することが出来ます。
5. 複写済みパラメータ： 右クリックメニューを使ってコピーされたオペレーターを表示します。
6. ランゲージ： オペレーターで使われるスクリプト言語をPythonとtscriptの二つから選択できます。
7. 拡張・折り畳み： 全てのパラメータの拡張・折り畳み表示を行います。
8. 非デフォルトパラメータ： デフォルト値から変更されたパラメータのみが表示されます。

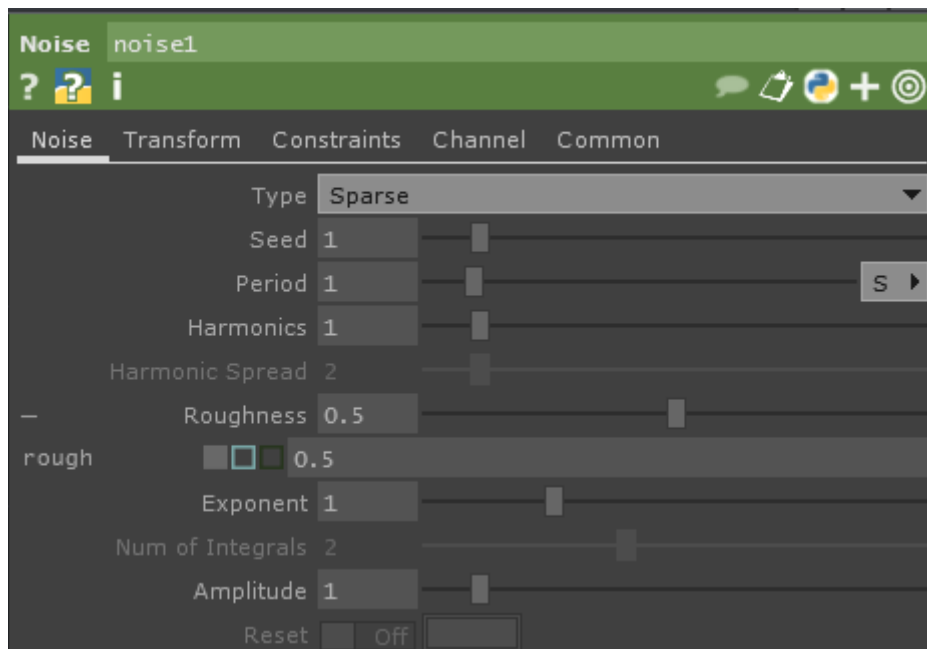
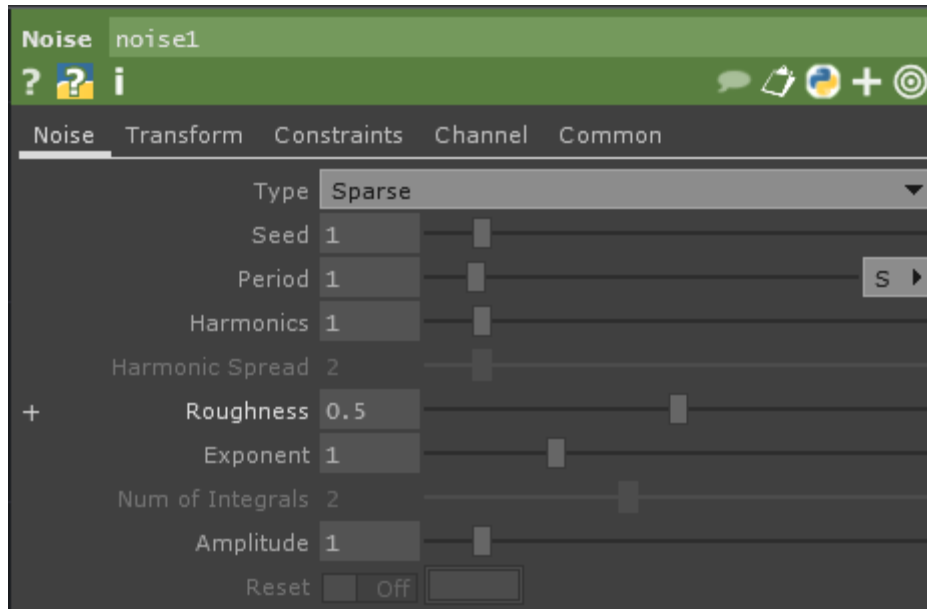
2.2 パラメータ

パラメータへの入力方法は様々なものがあり、状況によっては静的値(固定)である必要もあれば動的値(可変)が必要な場合もあります。パラメータには下記の三つのモードがあり、それぞれから違う効果が得られます。

1. Constant (コンスタント) モード
2. Expression (エクスプレッション) モード
3. Export (エクスポート) モード

ほとんどのパラメータのデフォルト設定はコンスタントモード(グレーの配色)になっています。Python、tscript又は数値入力にはエクスプレッションモードが使われます。エクスプレッションモードの場合は濃いグレーと薄い青の配色になります。エクスポートモードは直接CHOPチャンネルを参照する場合に使用されるもので、薄い緑に配色されます。

それぞれのパラメータには三つの内からどのモードでも適用することが出来ます。モードを変更するには、カーソルをパラメータの名前の上に移動し、カーソルが下の例の様に「+」に変わった時点で左クリックします。



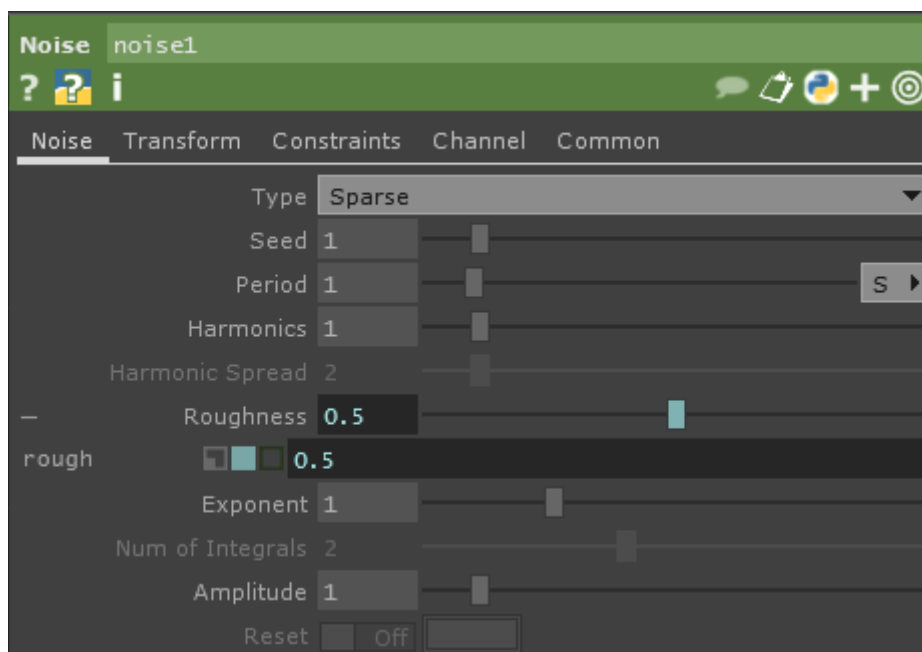
パラメータの名前を左クリックすると、下の様にパラメータが拡張表示されます。

パラメータが広がると、三つの項目が表示されます。一つ目は、一番左に表示されるパラメータのスクリプトネームです。上の例で説明すると、このNoise CHOPのRoughnessパラメータのスクリプトネームは「rough」になります。スクリプトネームはTouchDesigner内のあらゆるスクリ

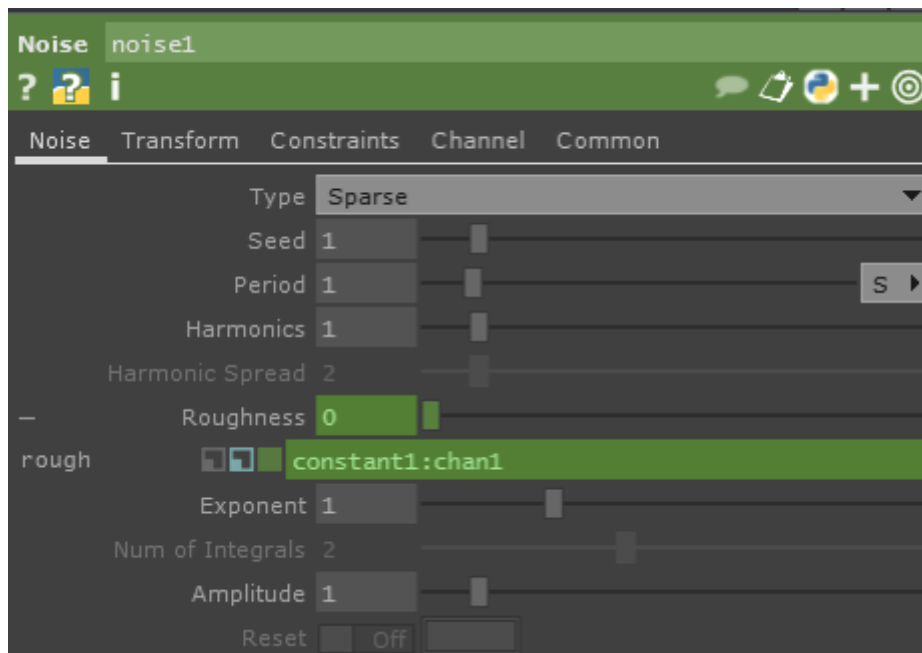
プト言語でこのパラメータが参照されるという重要な役目を果たします。更に、上の例のNoise CHOPのRoughness（荒さ）を「1」に設定するPythonスクリプトは次のようになります。

```
op('noise1').par.rough = 1
```

次に、パラメータのモードを示す小さな三つの正方形アイコンです。先程の説明の通り、デフォルト設定はコンスタントモードで、グレー一色に色づけられた正方形アイコンがこのモードに設定されていることを意味しています。薄い青に縁取られた正方形アイコンを左クリックすることにより、エクスプレッションモードへと変更できます。現在選択中モードの正方形アイコンは一色に塗りつぶされた状態へと変わり、そしてその右側のバリューフィールドも同じく選択中モードの色へと変わります。



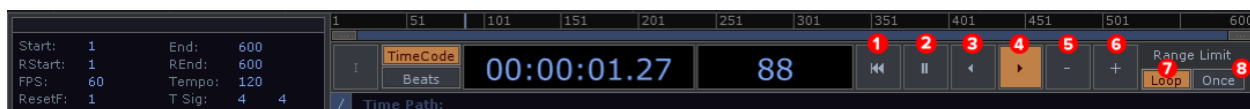
エクスポートモードに変更するには、CHOP channelを変更したいパラメータの上にドラッグ&ドロップして、「Export CHOP」を選択します。正方形アイコンが緑一色に塗り潰された状態へと変わること、エクスポートモードに変更したのを確認できます。



三つ目はバリューフィールドです。パラメータモードによってここに表示される値の種類は変わってきます。コンスタントモードでは現在の設定値が表示され、バリューフィールドを左クリックする事によって内容を編集できます。エクスプレッションモードでは評価対象であるPythonもしくはtscriptのスクリプトが表示されます。ここでも入力内容を左クリックする事によって編集が可能です。エクスポートモードではこのフィールドの内容は他のCHOPによって決定されるので、編集することはできません。コロン(:)を挟んで、前方にはこのパラメータにエクスポートしているCHOPへのパスが、そして後方にはそのCHOPによってエクスポートされるチャンネルの名前が表示されます。

2.3 トランスポートコントロール

使用方法と機能は他のソフトのトランスポートバーとはほぼ同じだと考えてもらっていいかと思います。トランスポートバーの機能を左から右へと簡単にご紹介しましょう。



1. タイムラインをフレーム1へとリセットします。
2. タイムラインを一時停止します。
3. タイムラインを逆再生します。
4. タイムラインを通常再生します。
5. 一つ前のフレームへと戻ります。
6. 次のフレームへと進みます。
7. 「Range Limit」で「Loop」を選択すると、タイムラインを継続的にループ再生します。
8. 「Range Limit」で「Once」を選択すると、タイムラインを一回再生した後、最後のフレームを表示した状態で停止します。

キーボードのスペースキーを使って、簡単にタイムラインを再生または一時停止することも出来ます。

2.4 タイムラインセッティング

メディアやアニメーションがタイムラインにロックされていない場合は、タイムラインセッティングを使用する必要はありません。タイムラインセッティングウィンドウは画面の左下に常に表示されています。ここではFPSとTempo設定が重要ポイントです。FPS (frames per second) とは一秒に何コマの静止画 (フレーム) がレンダリングされるかを意味します。デフォ

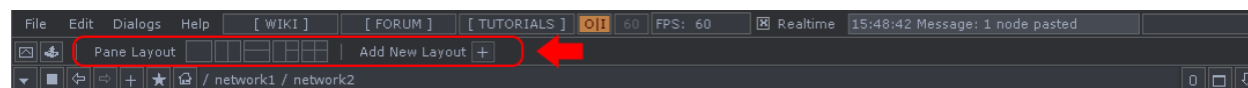
ルトで60FPSに設定されていて、これはTouchDesignerが一秒につき60フレームをレンダリングするという意味です。「Tempo」ではBeat CHOPのBPM (beats per minute) を設定することができます。

タイムラインセッティングを使用するのはほとんどがメディアやアニメーションをタイムラインにロックする必要がある場合です。「Start」と「End」はタイムラインの始まりと終わりのフレームを、「RStart」と「REnd」ではループ再生の始まりと終わりを設定できます。これらの設定を使えば、例えば4千フレームのタイムラインからなるアニメーションを構成し、調整や確認の為にタイムラインの一部のみをループ再生することも可能になります。

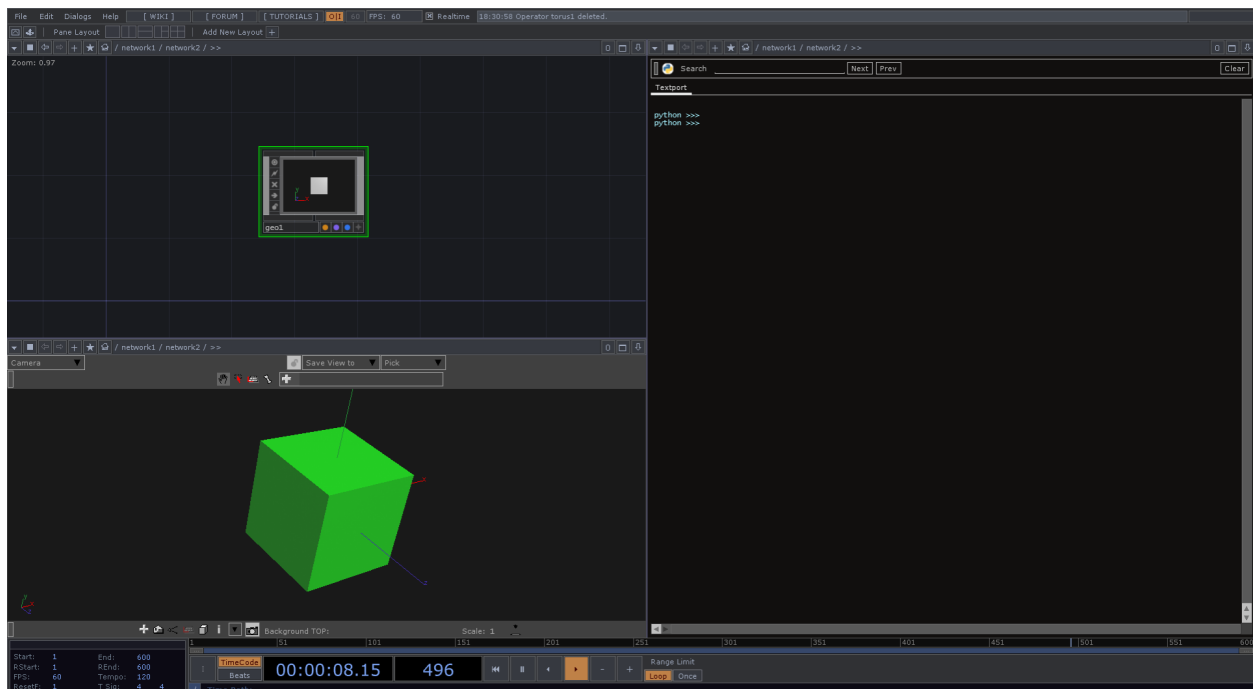
Start:	1	End:	600
RStart:	1	REnd:	600
FPS:	60	Tempo:	120
ResetF:	1	T Sig:	4 4

2.5 ペイン（作業領域）

ネットワーク間の移動にペインの使用を習慣付けると、更なる作業の効率化に繋がります。例えば、パラメータを調整する為に三つのネットワーク内を通して移動し、そしてその結果をチェックする為にまた元の場所に戻る・・・なんてことは時間の無駄に過ぎません。ペインを使えば現在使用中のウィンドウを縦または横に何分割にもすることができます。ペインのレイアウトは再び利用する為に保存することも出来ます。



既定では、ペインのレイアウトオプションは上のサンプル画面の様にプリセットされています。良く使われる、縦二分割、横二分割、三分割、及び四分割のレイアウトが用意されています。更に「New Layout +」ボタンを押して新規レイアウトの名前を入力し、「Create」をクリックすることによって、新しいレイアウトを作成・保存することもできます。分割サイズや位置を保存だけでなく、それぞれのペインの種類も保存されます。下の例を使ってどんなにユニークなレイアウトが可能かどうかをご説明しましょう。



ペインではダイアログボックス、ネットワーク、ビューアー等々のコンテンツを表示することができます。様々な種類の異なったコンテンツを一つの画面上に組み合わせることで更にフレキシブルで効果的な作業が可能です。上記の例では、左上のペインにはネットワークエディタが表示されています。その下にはジオメトリビューアーが、右側にはTextportが表示されています。沢山のレベルで構成されたプロジェクトでは上の例の様なペインレイアウトを使用することによって作業時間の短縮に繋がるはずです。上記のレイアウトを保存すると、先程の説明の通り、それぞれのペインの種類も保存されます。

ペインで使えるキー操作は次の通りです。

- **Alt** と **[**

Alt キーと左角括弧キーで、カーソルが位置するペインを縦に分割します。

- **Alt** と **]**

Alt キーと右角括弧キーで、カーソルが位置するペインを横に分割します。

- **Alt** と **Z**

カーソルの位置するペインを閉じます。

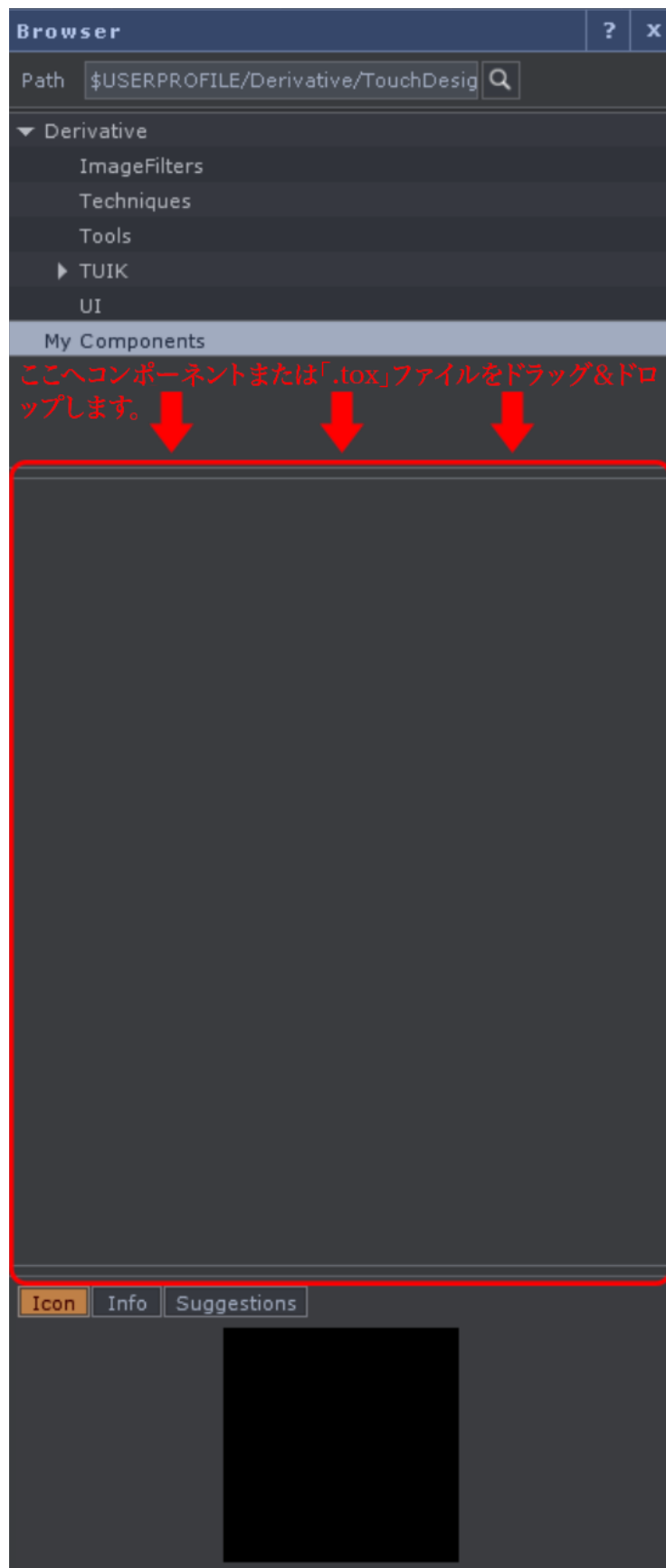
2.6 Palette Browser (パレットブラウザー)

パレットブラウザーとは、コンポーネント用ライブラリーとも言える物で、「.tox」ファイルの保管場所です。「.tox」ファイル内にはネットワーク及び複数のオペレーターをまとめたコンポーネントが保存されています。頻繁に使うオペレーター、UI、Pythonスクリプト等をコンポーネント内にまとめて「.tox」ファイルとして保存し、再度利用する際に簡単にアクセスすることが出来ます。

キーボードの「Alt」キーと「L」キーを押すとパレットブラウザーがフローティングウィンドウとして開きます。プロジェクトを新規で開く場合には、デフォルトでパレットブラウザーが画面左上にドッキングされた状態で開きます。パレットブラウザーを開いてデフォルトで用意されている「.tox」ファイルを見てみましょう。

「Derivative」セクションの下「Tools」を選択し、「blendModes」コンポーネントをドラッグして新規プロジェクト画面へドロップします。コンポーネントのUIの中には様々なコンテンツがあるのが見てとれるかと思います。まず始めに二つのTOPを作成して、TOP内の画像又は映像をブレンドする為に「blendModes」コンポーネントへと繋げてみましょう。「blendModes」コンポーネントの左下にある「+」フラグを左クリックしてビューアーをアクティブ化します。ボタンを押して好きなブレンドモードを選びます。もし「Crossfade」オプションを選択している場合には、ブレンドされたイメージ上にある半透明のバーを動かして二つの入力データの配合を調節できます。この様にとっても便利なツールをパレットブラウザーからドラッグ&ドロップするだけで使うことができるのです！

この本で目指しているゴールの一つは、新たなツールを制作して繰り返し利用する為にパレットブラウザーに保管することです。コンポーネントまたは「.tox」ファイルをパレットブラウザー内の「My Components」に保存するには二つの方法があります。「My Components」を左クリックし、ネットワーク内から保存したいコンポーネントをドラッグしてドロップエリア（下の例の赤い枠内）にドロップします。二つ目の方法として、ウィンドウズエクスプローラ内の「.tox」ファイルを同様にドラッグ&ドロップして保存することもできます。



2.7 検索ダイアログボックス

Search Dialog（検索ダイアログボックス）はプロジェクトが無数のPythonスクリプトとクモの巣のようなネットワークで複雑に構築されてくると欠かせない存在になります。検索ダイアログボックスは多様な項目を様々なロケーションから探し出すことができます。必要に応じてどんなに一般的で大まかな検索でも、特定の検索でも可能です。画面上部の「Edit」メニューから「Search」を選択するか、「F3」キーを押すことによって検索ダイアログボックスが開きます。

「Basic」（基本検索）ではオペレーターとPythonスクリプトが検索対象です。Pythonスクリプトはオペレーターのパラメータを動的に変化させる為に良く使われますので、スクリプトが増えるに連れて、どのスクリプトをどこで使ったか全てを把握するのは難しくなります。検索の例として、下記のキーワードを検索すると、名前に「transform」を含むオペレーターのパラメータを変化させるスクリプトの全てが検索結果として表示されます。

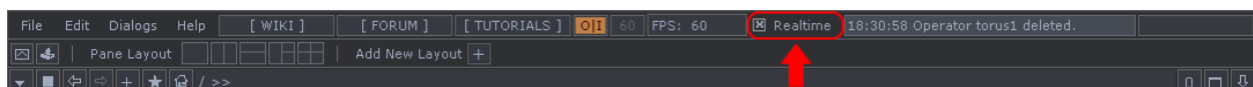
`op('transform').par`

沢山のスクリプトが存在するネットワーク内を手作業で探して回るより、はるかに簡単にキーワードを見つけることができます。

「Advanced」（詳細検索）ではオペレーターの名前、種類、コメント、フラグ等を指定して検索することができます。過去に制作したプロジェクトを再び使用する場合（特に複雑な物の場合）は、詳細を再度把握したり特定のオペレーターを見つけるのに長時間かかることが多々ありますが、そんな時には詳細検索が便利な存在だと言えます。例えば、名前に「movie」のキーワードを含んだ「Move In TOP」を検索するとします。この場合はオペレーター名の一部とオペレーターの種類を指定した検索を行うわけですが、この様なシンプルな検索でも条件の一致するオペレーター全てがリストにされて表示されますので十分に詳細な検索結果が得られます。

検索結果をダブルクリックすることによって新しいウィンドウが開き、検索されたスクリプトの位置するネットワークが表示されます。現在の作業を妨げることはなく、新しく開かれたウィンドウ内で編集を行うこともできます。

2.8 リアルタイムモード



画面上部にあるRealtime（リアルタイム）フラグがオンになっているかどうかによってタイムラインがどの様に再生されるかが大幅に変化します。オンになっている時（デフォルトではオンです）、システムはフレーム数よりも時間を優先して再生します。例えば、30秒のムービーファ

イルならば、TouchDesignerはフレーム数を排除してでも、ちょうど30秒で再生するのです。ほとんどのリアルタイムインストレーションやパフォーマンスにはこのモードが使用されます。

リアルタイムフラグがオフになっている時（フラグが赤くなった状態）ではシステムはフレームレンダリングを優先して再生します。先程の例で、もしリアルタイムフラグがオフになっているとしたら、例え再生時間が30秒より長くなるとしても、TouchDesignerは一つ一つ全てのフレームを処理してレンダリングするのです。このモードは複雑な構成のアニメーションや3Dレンダリングを外部保存する際に便利です。

2.9 便利なキー操作

TouchDesignerで重宝するキー操作をいくつかご紹介しましょう。

1) カーソルがネットワークエディター上に位置している状態でのキー操作

- 「p」キー 現在使用中のオペレーターのパラメータウィンドウを開け閉めします。
- 「o」キー ネットワークの全体図（画面の左下に表示）を開け閉めします。
- 「c」キー カラーパレットを開け閉めします。選択中オペレータの枠の色を変えることが出来ます。

2) オペレーターを選択した状態でのキー操作

- 「a」キー オペレーターのビューアーがアクティブ化されます。
- 「b」キー バイパスフラグのオン・オフ切り替えをします。
- 「h」キー 右クリックメニューの「Home All」と同様に、ネットワーク内全てのオペレーターを画面上に納めて表示します。
- 「r」キー Render Flag（レンダリングフラグ）のオン・オフ切り替えをします（SOP、Object COMPの場合のみ）。
- 「d」キー Display Flag（ディスプレイフラグ）のオン・オフ切り替えをします（適用オペレーターの場合のみ）。
- 「Control」 + 「c」 選択中のオペレーターをコピーします。
- 「Control」 + 「v」 コピーされたオペレーターを貼り付けます。
- 「Control」 + 「Shift」 + 「v」 カーソルの位置にコピーされたオペレーターを貼り付けます。

3 TOP

3.1 はじめに

テクスチャーオペレーター（TOP）はTouchDesignerで作成するプロジェクトにおいて欠かせない存在です。2Dのテクスチャーオペレーターであり、映像再生から3Dレンダリング、合成、ハードウェアのビデオ入力・出力までこなします。そして、モニター、プロジェクター、LEDディスプレイ等に出力する全ての工程で使われます。

3.2 Movie In TOP

数々のTOPが存在しますが、Movie In TOPはその中でも利用頻度の高いものです。

TouchDesignerに、静止画から様々なコーデックの映像ファイルまで幅広い種類のアセットを読み込むことができます。Movie In TOPで良く利用されるファイルフォーマットの一部をこちらにご紹介します。

- .mov
- .mp4
- .avi
- .tiff
- .jpeg
- .png

その他の読み込み可能なファイルフォーマットはDerivative社のTouchDesignerウィキページの「File Type」の欄でご参照下さい。

Movie In TOPはフレームレートの異なったアセットの取り込み、または出力の際の問題を軽減してくれます。Movie In TOPは、常時アセットの再生時間に従って再生します。例えば、プロジェクト自体が60FPSに設定されていて、30FPSで作成された10秒間のアセットを再生するとします。Movie In TOPはFPSの違いには関係なく必ず10秒間で再生するのです。逆の場合でも同様に、例えば60FPSで作成された10秒間のアセットを30FPSのタイムラインで再生したとしても、必ず10秒間で再生されます。どちらの場合にしても、10秒間というアセットの再生時間を優先するためにフレーム数を倍にする、もしくは半分削減して調整します。場合によってはフレーム補間を行うことによって、よりスムーズに再生できます。

3.3 ムービーファイルのプリロード（事前読み込み）

作成したリアルタイムアプリケーションがコマ落ちし続けると、伝えたいメッセージや魅せたい効果が台無しになってしまいます。再生時の問題の解決策は後のチャプターで詳細にご説明しますが、どの様な予防措置があるかをここで紹介しましょう。Movie In TOPのプリロード・

アンロードを使用することで、コマ落ちを予防することができます。この方法はとても簡単ですが、スクリプト言語を使用するために初心者ユーザーには敬遠されがちな方法でもあります。

「Preloading.toe」のサンプルファイルを開くと三つのボタンが画面上に現われます。「Preload」ボタンを押すと下記のPythonスクリプトを使って「moviein1」オペレーターの「Tune」タブ内にある「Pre-Read frames」パラメータに設定されたフレーム数がメモリにプリロードされます。

```
op('moviein1').preload()
```

「Play」ボタンでMovie In TOPを再生します。そして、「Unload」ボタンを押すと「moviein1」の再生が止まり、Movie In TOPで使われているメモリを解放（アンロード）します。この動作は下記のPythonを使って行われます。

```
op('play').click(0)
```

```
op('moviein1').unload()
```

動画を再生する際には、コマ落ちを防ぐ為にこの様に事前にプリロードすることをお勧めします。

3.4 Null TOP と Select TOP

Blur TOP等、メモリの大きな容量を要するオペレーターに比べて、Null TOP や Select TOP等はメモリへの負担は少ないので、どんどん利用して頂きたいと思います！この二種類のTOPはデータに対して何の変化も与えないものの、ワークフローを効率的にするのに大変便利な存在です。

ワイヤーでごちゃごちゃになったネットワークと、きれいに整頓されたネットワークとの違いは、Null TOP と Select TOPの使用法です。サンプルファイル「Null_1.toe」と「Null_2.toe」を開いて見てください。一つ目のファイルでは沢山のTOPが一つに集約されているのが見て取れます。この例では沢山のワイヤーが四方八方に伸びて、オペレーターとワイヤーが重なり合い、データの流れを追うのが難しい状況です。「Null_2.toe」では、最終の合流ポイントの前に、縦に並んだNull TOPがワイヤーを一行に整頓しています。この一行のNull TOPがチェックポイントとしての役目を果たし、一目見るだけでも数々のデータの流れを追うのがとても簡単になります。

Select TOPもデータに変化を与えませんが、ワイヤーなしで他のTOPのデータを参照・使用することが出来る為、ネットワークを整頓するのに大変役立ちます。サンプルファイル

「Select_1.toe」を開いてください。この例ではIn TOPとOut TOPを使用してコンテナ間のワイヤーを引き込んだ際のごちゃごちゃとした状態が見て取れます。この例では、12個のアイテム

を繋げているだけですが、もしこれが100のアイテムを必要とするプロジェクトだとしたら、一体どのような状態になってしまうのでしょうか！？Select TOPはそんな時に便利な存在です。

「Select_2.toe」のサンプルファイルを開いて見てください。急に増えたアイテムの数にも関わらず、ごちゃごちゃに繋がったワイヤーもなく理解しやすくなっています。「composite」コンポーネント内にあるSelect TOPで作成された画期的な参照システムを見てみましょう。この方法は先程の例の様に手作業でワイヤーを繋げた場合より遥かに効率的で、それぞれのSelect TOP (item 0から36まで) の「TOP」パラメータ内のPythonスクリプトが一レベル上のネットワークから、item番号の一致する複製TOPのコンテンツを参照して取り込みます。更に詳しく説明すると、Select DAT (/project1/composite/select2) は一レベル上のネットワークでReplicator を操作しているTable DAT (table1) と自動的に同期されています。Select DAT (/project1/composite/select2) が操作されることによってReplicator COMP (/project1/composite/replicator1) が下にSelect TOPを追加します。新規アイテムが一つ目のReplicator COMP (/project1/replicator1) に追加される度にComposite TOP (/project1/composite/comp1) とワイヤーで繋がれたSelect TOPが追加されるのです。

現時点でこのコンセプト全てを理解するのは難しいかと思いますが、ご心配なく。Replicatorとスクリプトの使用方法については後程ゆっくりとご紹介します。今理解して頂きたいのは、Select TOPとシンプルなスクリプトを使用した事によって、このプロジェクトにはこれからメンテナンスがほとんど必要なくなった、という点です。他のアイテムを複製したい場合にはテーブルに一行加えるだけで可能になったのです。

3.5 コーデック

作成したムービーファイルをスムーズに再生するのは決して簡単な作業ではありません。使用するコーデックによって再生時のビジュアルの質は大きく左右されるので、様々な種類を試して、作成したプロジェクトに合うコーデックを選出するのをお勧めします。

コーデックの種類の説明の前に、コーデックとコンテナの違いを理解するのが大切です。一般的に、コーデックが音声・映像ファイルの形式（フォーマット）を意味する言語として理解されがちですが、コンテナの種類によってはコンテナ内にコーデックを格納することもありますので、ビギナーにとっては分かりにくい場合が多いです。

Codec（コーデック）とはcompressor-decompressor（圧縮・解凍を行うもの）を短縮した言葉で、その意味の通り二つの役目を果たします。一つ目は保管と移動の為に映像データの容量を圧縮すること、そして二つ目は再生の為に解凍することです。この様に異なった二つの機能がある為に、数ある種類の中から目的に合ったコーデックを選ぶ必要があります。容量が軽く、移動性の高いファイルの圧縮に向けたコーデックがあれば、質の高いコンテンツの長期保管向

きのコーデック等、様々な種類があります。プロジェクトによって必要な種類は変わってくるでしょう。例えば、一つのコンテンツを最良の画質で再生したい場合もあれば、複数のコンテンツを同時再生する為に低い画質で再生することが必要な場合もあるでしょう。プロジェクトに合ったコーデックを選出するにはリサーチやテストが必要ですが、その努力が後々素晴らしい結果に繋がります。

コンテナはその名の通り、圧縮されたビデオや音声、そしてそれらのコンテンツを解凍・再生する為に必要な全てのメタデータを格納します。コンテナの種類も複数ありますが、コーデックに比べると、どのコンテナを選ぶかによってプロジェクトに与える影響は少ないと言えます。

様々な種類のコーデックとコンテナを組み合わせる場合は更に複雑です。例えば、「test_movie.mov」というファイルが存在するとします。このファイルは「.mov」 QuickTimeコンテナ内のAnimation Codecで圧縮されたビデオファイルである可能性もあれば、同じくQuickTimeコンテナ内のH.264コーデックで圧縮されたビデオである可能性もあるのです。更に例を挙げると、同じH.264コーデックの圧縮ビデオは「.mp4」もしくはMPEG-4 Part 14コンテナ内に格納されていてもおかしくはありません。この様に多様な組み合わせの可能性が面白い点でもあり、ビギナーにとっては悩ましい点でもあります。

現在よく使われているコーデックはHAPシリーズ、H.264、Animation Codec、そしてCineformです。それぞれのコーデックには長所と短所がありますので、下記のリストで簡単にご説明しましょう。

HAPシリーズ

メリット

- 極めて高いフレームレートそして良い画質での再生が可能
- 最小限のCPUへの負担
- 最小限のGPUへの負担
- ロスレス圧縮

デメリット

- ファイルサイズが大きい
- Windowsでのバッチエンコードが困難
- 再生時に SSDもしくはSSDによるRAIDが必須
- 読み込み速度の速いハードドライブが必須

H.264

メリット

- ファイルを極めて小さいサイズに圧縮可能
- 高い圧縮率でありながらも高画質（同レベルの圧縮率のものに比べても最良の画質）
- ストレージ容量を節約できる

デメリット

- 高解像度または高フレームレートの再生にはCPUの性能に左右される
- 適切にエンコーディングがされてない場合は、色が正しく表示されない（クオンタイズされる）可能性あり
- ビットレートがコンテンツによって大きく異なる
- 4096×4096ピクセルのサイズ制限
- アルファチャンネルに非対応

Animation Codec

メリット

- クオリティを100%に設定すればロスレス圧縮が可能
- 圧縮のクオリティが最優先される
- アルファチャンネルに対応

デメリット

- 大容量のファイルサイズ
- ハードドライブとCPUへの負担が大きい
- 複雑なコンテンツの場合はビットレートが激しく変動

Cineform

メリット

- 一定のビットレート
- 高画質
- アルファチャンネルに対応

デメリット

- 大容量のファイルサイズ
- Cineformのエンコーディングソフトの購入が必須

4 CHOP

4.1 はじめに

CHOP（チャンネルオペレーター）はチャンネルに関連する全ての作業を扱うオペレーターです。モーションデータ、音声の入力、アニメのキーフレーム、DMX、MIDI、OSC、そして Microsoft Kinect、Leap Motion、Oculus Rift、ペンタブレット、キーボード、マウス等のハードウェアからの入力の作業を行います。そして、下にリストアップされている様々なタイプの音響・映像機器とのデータ通信（入力、処理、出力）を行います。

- ミキサー
- MIDIコントローラ
- シンセサイザー
- DMXの照明機器
- Microsoft Kinectカメラ
- TouchDesignerが動作しているコンピュータ
- スピーカー
- その他の音響・映像アプリケーション（Max/MSP、Ableton Live、Resolume Arena等）

4.2 データの送受信

MIDI通信には様々な種類のソフトウェアとハードウェアを使用することができます。MIDIの入力・出力はAbleton Live、Avid Pro Tools、Propellerhead Reason等のDAW（Digital Audio Workstation）を使用して、手早く簡単に行える作業です。オーディオコントローラには通常MIDIが搭載され、USBケーブルを使って通信することができます。ボタン、フェーダー、ピアノの鍵盤、タッチストリップ、ジョグホイール、ドラムパッド、ポテンシオメーター等の入力ハードウェアもコントローラの一部です。

Cycling 74 Max/MSP、PureData、Native Instruments Reaktor等の開発環境ではOSC通信が可能です。OSC通信には構造上の改良により、モダンなネットワーク技術、MIDIよりも高度なレベルでの通信、チャンネルに名称をつけられる機能等の様々な利点があります。OSC通信はUDPまたはTCPを使って行われる為、アプリケーション間の連携や長距離間のリアルタイムのメッセージ送信を簡単に行うことができます。現在ではソフトウェアとコンピュータシステム間の通信手段の定番として使用されています。

DMXは照明機器や照明コントローラの為の通信プロトコルです。DMX対応機器にはディマー、チェイス機能、RGBチャンネル、モーターによる駆動等の為に数々のチャンネルが用意されています。多くの照明コントローラはDMXプロトコルを使用して、ビデオコンピュータシステム

や照明機器と通信します。数多くのコントローラが市場にあり、使い方は様々です。照明演出を使ったプロジェクトを制作する場合には、それぞれの操作マニュアルを良く理解するのが重要です。例えば必要ないとしても、全てのチャンネルを使用前に設定しておくことが必須です。DMXデータの送受信を最適化する為のチャンネルの整理・管理法は後程紹介いたします。

Sync In CHOPとSync Out CHOPを使ってTouchDesignerシステム内、またはシステム外の複数のセッションを同期することができます。データの送受信にはOSCプロトコルが使用されます。この2種類のCHOPは全ての同期された機器のフレームの状況を常時把握して通信します。全ての機器で特定のフレームがレンダリングされたのを確認すると同時に次のフレームへと進み、この作業が繰り返して同期されている機器全てが常時同じフレームを映し出すのです。

4.3 音声の入力・出力

音声は様々なソースからの取り込みが可能で、そして様々な処理方法があります。TouchDesignerを使って音声ファイル、動画ファイル、ネット上のストリーミング音声、外付け音声インターフェース等の音声を処理することが可能です。そして、TouchDesigner自身で合成して一から音声を作り上げることもできます。

サウンドデザインと音声トラックのプロジェクトには必ず専用の音声ファイルがあり、TouchDesignerはAudio File In CHOPとAudio Play CHOPを使ってMP3、AIFF、WAV等の広く使われているファイルフォーマットを読み込み・再生することができます。リピート再生、設定ポイントへのスキップ、ピッチ変更、カット編集等、音声ファイルをフレキシブルに扱うことができます。

Audio Movie CHOPは動画ファイルの音声のみを再生したい時に使用します。ファイルを参照する代わりにMovie In TOPを参照します。音声を再生するのに参照したMovie In TOP内の映像と同期して再生する他、音声オフセットパラメータで同期タイミングをコントロールすることができます。

色々な種類の外付け音声インターフェイスをTouchDesignerで使うことができます。Derivative社のTouchDesignerフォーラムとWikiページにて詳細に対応デバイスをリストアップしていますのでご覧ください。

これらの対応デバイスはアナログもしくはデジタルの音声データを出力・入力します。入力ソースは音楽家・楽器家、オーディオミキシングコンソール、プロのビデオカメラ、そしてその他のコンピュータシステム等様々です。音声データの出力先は以前に紹介した様々な種類の機器（スピーカーを含む）です。Audio Device In CHOPとAudio Device Out CHOPを使用して外付けインターフェイスとの通信を行うことができます。先方はTouchDesignerへの入力を、後方は

TouchDesignerからの出力を行います。そして、外付けのSDIソースからNVIDIA Quadro SDIカードを使用して音声データを入力する場合には、Audio SDI CHOPを使用します。

TouchDesignerで利用できる音声ドライバーには二つの種類があります。DirectSoundはDirectXの一部品であり、TouchDesignerの以前のバージョンでも使われてきました。長期間に渡って開発された成熟ドライバーで、CPU使用率が高い場合でも比較的遅延時間が少ないのが特徴です。

ASIOはTouchDesigner 088で新しく対応可能になったドライバーです。Steinberg社によって開発されたドライバーで、「WindowsのOSのみを介しての音声通信」というDirectXの大きな欠点を改善したものです。OSを経由せずに外付けハードウェアと直接通信することによって、DirectSoundよりも更に低いレイテンシー（遅延時間）が可能となりました。

TouchDesignerで入力と出力音声データのセットアップが完了すれば、他のデータと同様に自由に扱うことができます。

4.4 サンプルレート

ほとんどのアプリケーションではバックグラウンドでのプロセスの過程を詳細に目にすることが少ない為、オーディオを数学的に処理することに慣れているユーザーは少ないと思います。実はオーディオとはとてつもなく速いスピードで処理されている数字データなのです。このコンセプトを理解すれば、TouchDesignerでのオーディオの扱い方に役立つはずです。

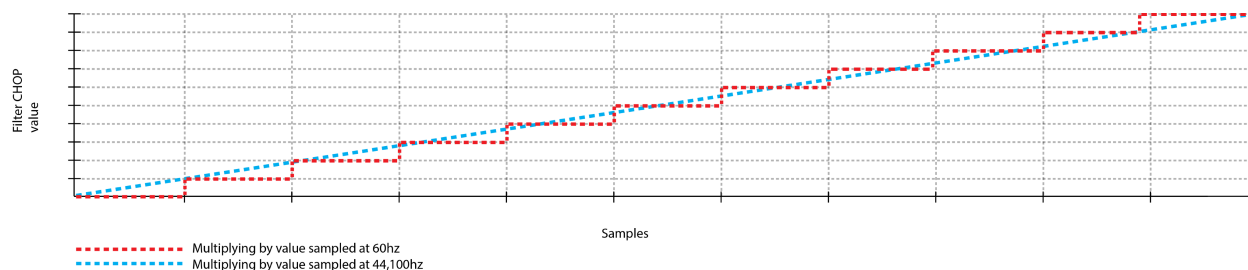
「Sample_rates_1.toe」のサンプルファイルを開いてみて下さい。ここではオーディオアプリケーションで最も基本的な機能の一つである「ミュート」機能を作成します。これは、Math CHOPを使って、音声データにButton COMP内の出力値（0または1）を掛けることによって可能になります。通常の数学の計算と同様に、0を掛けた値（この場合は音声データ）は必ず0になります。同様に、値が何であろうと、1を掛ければ変化なくそのままの状態を保ちます。この二つの出力値を使って音声をオン・オフすることができます。

更にフェードイン・アウト機能を追加して詳しくご説明しましょう。「Sample_rates_2.toe」をご覧ください。

先程の例に更に二つのオペレーター、Filter CHOPとResample CHOPが追加されています。Filter CHOPは入力値に滑らかさを与える物で、オン・オフ二つの異なる出力値を滑らかに繋ぎます。

初心者ユーザーは他のCHOP内のサンプリングレートを見過ごしがちですが、アーチファクトのない音声再生には重要な物ですので十分に注意を払う必要があります。Oscillator CHOPは毎秒44,100回サンプリングされていますが、Filter CHOPは毎秒60回でサンプリングされています。これでは、出力値を掛けた時に音声サンプルとランプサンプルが1対1ではなく、735対1の比率に

なってしまう、二つを掛け合わせた時に、735のサンプルごとにピッチが上がった、もしくは下がった状態で再生されてしまいます。下の図の青の点線は1対1、赤の点線は735対1の比率を示しています。



上の図では、異なった二つのサンプリングレートを掛け合わせた時に起こる階段状のガタガタとした状態が見て取れます。こういった場合に、Resample CHOPはサンプリングレートを一致させる役目を果たします。ほとんどのCHOPではプロジェクトFPSが規定のサンプルレートとして設定されていますので、プロジェクトFPSが30に設定されている場合は更にこの現象がひどくなります。先程の例を使うと、レートが30の場合は、音声サンプルとランプサンプルの比率は更に広がり、1470対1になってしまいます。この場合、なんと1470サンプル毎にしかボリュームが変わらないという事になってしまいます！

この様に、それぞれのCHOPのサンプリングレートを把握しておくのは大変重要なことです。コントロールデータを異なるレートで入力・出力しなければいけない場合もありますので、状況に応じて対応しましょう。サンプリングレートを調節する必要がある場合はResample CHOPを使用しましょう。

4.5 タイムスライシング

タイムスライシングはTouchDesigner独特の機能で、慣れるまで少々時間がかかるコンセプトかと思います。

タイムスライスとは最後にレンダリングされたフレームと現在レンダリングされているフレームの間の時間です。常に変化を遂げている時間の間隔だと考えて貰えばいいかと思います。もしプロジェクトが常に60FPSで機能している場合にはタイムスライスは1フレーム間だという事になります。逆に、実時間についていけず10フレームをドロップした場合には、タイムスライスは10フレーム間になります。

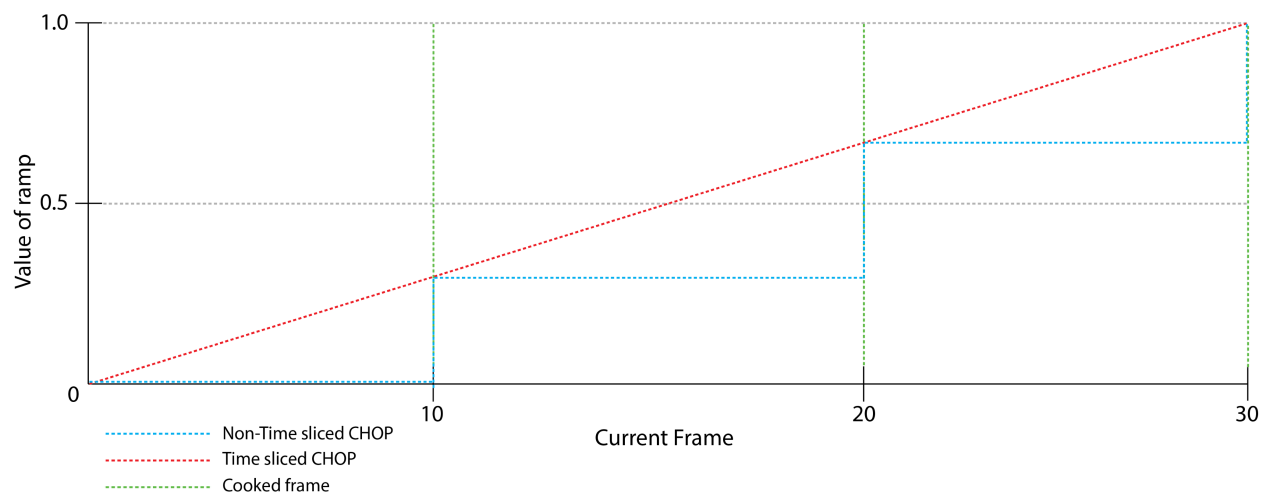
タイムスライス機能は、CHOPのみで使える機能であり、コマ落ちが生じた際にデータを滑らかに仕上げることができます。CHOPが、落ちたフレームに気づき、計算に入れ始めた時にタイムスライスが生じます。簡単に言えば、臨機応変なレンダリング機能なのです。タイムスライスの

時間が長くなるに連れて、CHOPは落ちたフレームを補い、滑らかなアウトプットを制作する為に必要な量のフレームを自動的にレンダリングするのです。タイムスライス機能がOFFになっているオペレーターでは、例えばいくつものフレームが落ちたとしても、処理される値が一つ前のフレームでレンダリングされ、そしてまた次のフレームの値へとスキップしてしまいます。

タイムラインが常時30FPSでプロセスしている場合はタイムスライスは常に1フレーム間です。そして、0から1まで1秒間（30フレーム）の間に到達する二つのカーブがあるとしたら、二つとも滑らかなカーブになります。もし、何らかの原因で10フレーム中の1フレームしかレンダリングされなかったとしたら、全く異なる結果となります。

タイムスライス機能がOFFになっているCHOPでは、処理される値がスキップして、レンダリングされたフレーム間のデータは失われてしまいます。逆に、タイムスライス機能がONになっているCHOPでは、最後のフレームと現在のフレームの間のデータを補う為にドロップされたフレームをレンダリングします。この様に、何が起きようとデータを滑らかに保つことができるのです。

下の図は先程の例の二つのカーブを表しています。青の線がタイムスライス機能がOFFになっているCHOP、赤の線がタイムスライス機能がONになっているCHOPを、そして緑の線がレンダリングされたフレームを示しています。



5 DAT

5.1 はじめに

DAT（データオペレーター）はデータに関する作業を行います。DATを使って様々な形式のデータを作成、編集、解析、送信、そして受信することができます。テキスト、テーブル、Pythonスクリプト、XML、JSON、MIDI、Serial、OSC、そして他の様々なデータ形式に対応しています。

DATとPythonスクリプトはTouchDesignerによって構築されたロジカルシステムにとって、なくてはならない存在です。様々な情報やメタデータで埋め尽くされたテーブルを解析したり、他のオペレータの状態を監視したり、他のシステムからの信号に従って操作を行ったりと、様々な機能によってTouchDesignerは複雑なシステムを作り出すことができます。これらはTouchDesignerの特徴と言える機能なので、いくつかのロジカルシステムの例を後のサンプルチャプターで紹介しています。

TouchDesignerを柔軟性のあるPythonプログラミング環境だと考えればより良く理解して頂けるはずです。複雑で長文のPythonプログラムをフレキシブルで使用しやすい小容量の機能に振り分ける事が可能だと考えて見て下さい。まさにその分割された機能がText DATなのです。この機能は簡単に設定、管理、追加が可能な上に、手軽にシェアやコラボレーションをすることもできます。

5.2 通信方法

DATの数々は一般的な通信プロトコルを使ってデータを入力・出力することができます。

TouchDesignerはMIDI、OSC、TCP、UDP、UDT、そしてWebsocketを使ってネイティブに通信することができます。その結果、様々な種類のプログラム、ウェブアプリケーション、ウェブサービス、舞台照明コントローラー等と通信することができるのです。

MIDI、OSC、DMXプロトコルについては第4章（CHOP）で説明しましたが、DATではそれ以上に様々なプロトコルを使用しての通信が可能です。

TCPはインターネット上で標準的に用いられる通信プロトコルです。TCPはコネクション型プロトコルであり、明確なクライアント・サーバー関係が存在し、そして両者間のコネクションが確立（通常はバックグラウンドで）されてから初めてデータ送信が始まります。この関係のおかげでTCPコネクションは信頼性が高く、そしてクライアント・サーバー両者が互いにデータの送受信を確認することが可能になるのです。その為、データが喪失されることはありません。

TCPは順番付けされた送受信ストリームであり、データが送信された順番通りに相手に受信されます。

逆に、UDPはコネクションレス型のプロトコルで、データの送受信前に明確なコネクションの確立はされないため、送受信確認、パケット喪失、順序制御といった点では信頼性が低いプロトコルです。これらの特徴が不利にならない（データの喪失よりもパケットの遅れの方が問題になる）プロジェクトでは、UDPを使用することによってパフォーマンスの向上に繋がります。

そして、TCPとUDP 両方の利点を生かして制作された新しいプロトコルがUDTです。UDTはUDPベースですが、コネクション型で信頼性の高いプロトコルです。TCP同様の送受信確認と順序制御機能を備えていると同時に、UDPのスピードを有するプロトコルです。

コネクションレス型のメリットは「マルチキャスト通信」機能だと言えるでしょう。マルチキャスト通信とは、同じ内容のメッセージを、一回のみの送信動作で、ネットワーク上の全てのコンピュータに送ることができる、という意味です。一つ一つのコンピュータに同じメッセージを送信する必要がないので、負荷は最小限に収まります。「ユニキャスト通信」は対照的な通信方法で、例えば同じ内容のメッセージを複数のコンピュータに送信するとしても、一人の受信者に対して一つずつメッセージを送信する必要があります。

ウェブブラウザ及びリアルタイムウェブアプリケーションとの通信の場合はWebsocketが主に利用される通信方法です。Websocketは制作過程から徹底してウェブブラウザとサーバー間の通信用に作られたプロトコルで、効率的な双方向通信を可能にしました。

6 SOPs

6.1 はじめに

SOPはサーフェス（表面）オペレーターの略で、色々な種類があります。SOPは3Dに関する全て（三次元形状、パーティクル、アーキテクチャモデル、3Dキャラクター等）の作業を行うものです。ビギナーユーザーがSOPに注目してわざわざ学ぼうとする事は少ないですが、SOPを熟知する事によってTouchDesignerプロジェクトの限らない可能性と効率的な問題解決方法を発見できるはずです。

SOPなしでは、プロジェクションマッピング、リアルタイム三次元モーションキャプチャ、建物へのLEDファサードのインスタレーション、マルチレイヤービデオの再生等はとてつもなく複雑でほとんど不可能だと言えます。

TouchDesigner088では現在、下記のタイプの3Dファイルに対応しています。

- .fbx
- .obj
- .3ds
- .dxf
- .dae

スムーズな再生の鍵は、オペレーターの不必要な計算を防ぐことです。その点については後程、「Optimization」の欄で詳しく説明いたしますが、SOP使用時には更にこの点が重要となります。トランスフォームアニメーションデータを直接SOPに適用するのではなく、常にGeometry COMPに適用するようにしましょう。何故ならSOPの変換はCPU内で行われ、そしてジオメトリ内の全ての頂点で実行されるからです。それに比べて、コンポーネントでの変換は対象ジオメトリ全体に適用され、GPU内で一つの動作として処理されます。GPUでの一つの動作とCPUでの何百もしくは何千という動作に比べると遥かにスムーズな処理となるはずです。

モデルの種類によって、点、プリミティブ、頂点、そしてメッシュの総数は変わりますが、基本的には多くのポリゴンや点があるほど大容量のメモリが必要になります。TouchDesignerには複雑なモデルのポリゴン数を減らすツールがありますが、専用ソフトウェアを使用してジオメトリを最適化する方が、フレキシブルな対応ができて良いでしょう。

6.2 レンダリング

ビギナーにとって難しいのは3Dワークフローから2Dワークフローへのスピーディーかつ効率的な転換だと言えるでしょう。インターネット上で3Dレンダリングについての様々な解説を読む

ことができますが、このチャプターの目的はシンプルな3DオブジェクトをRender TOPでレンダリングする事です。

3D画像を構成する三つの要素は次の通りです。

1. 3Dジオメトリ（及びマテリアル）
2. カメラ
3. ライト（照明）

「Rendering_1.toe」サンプルファイルを開いてみて下さい。シーンをレンダリングする為に必要な、基本の三つのアイテム（3Dジオメトリ、カメラ、ライト）がRender TOPに参照されています。始めにレンダリングの設定を詳細に紹介してから全体的なコンセプトを説明したいと思います。

3Dジオメトリは全ての中心とも言える存在なので、最初にご説明しましょう。3Dモデルは、ポリゴンだけで構成されているシンプルな物もあれば、アニメの3Dキャラであったり、建築物のモデルであったりと、様々です。既存のモデルをインポートする場合でも、初めから作成する場合でも、全ての作業はSOPによって行われ、そしてGeometry Componentによってレンダリングされます。ここで理解して頂きたいキーポイントは「直接SOPがレンダリングされるのではなく、レンダリングフラグがオンになっているSOPを含んでいるGeometry ComponentもしくはGeometry COMPがレンダリングされる」という事です。これを理解するのはTouchDesignerを使用する上で大変重要です。次の二つの例を使って詳しく説明しましょう。

「Rendering_2.toe」のサンプルファイルを開くと、異なったTranslateパラメーター値でレンダリングされている四つのGeometry COMPと、そのそれぞれへデータを送る一つのSOPが見て取れます。ここでは例としてBox SOPを使っていますが、実際には更に複雑なモデルの場合も良くあるでしょう。例えば、複雑なキャラを動作させる場合、数回反復するだけでシステムの許容範囲を超えてしまう事があります。この例の様に、一つのSOPを使用する事により、データの読み込みは一回のみとなり、Geometry COMPでのレンダリングや反復が可能になります。

サンプルファイル「Rendering_3.toe」では前の例とは異なり、一つのGeometry COMPの中にレンダリングされている三つのモデルがあります。レンダリングの対象になるのはGeometry COMPなので、その中に全てのモデルをまとめるとプロジェクトの効率性が上がります。この二つの例の違いは現時点では定かではないかもしれませんが、プロジェクトが複雑になるにつれて、効率性を維持することで作業時間の削減に繋がるでしょう。

これまで紹介したジオメトリはマテリアル（素材）を何も使用していないものばかりです。マテリアルは3Dシーンを更に奥深くするもので、例えばただの3Dの箱をセメントのレンガの様に仕上げるのもマテリアルです。マテリアルはSOPレベル、そしてコンポーネントレベルの二段階

で適用することができます。一つ目はMaterial SOPを使用して、そして二つ目はGeometry COMPのRenderタブのマテリアルを参照することで可能になります。「Rendering_4.toe」のサンプルファイルでは二つのどちらのモデルも同じマテリアルを使用していますが、それぞれが先程の異なった二つの方法でマテリアルを追加しています。

マテリアルに続いて、次はUVテクスチャマップです。サンプルファイル「Rendering_5.toe」は先程の「Rendering_4.toe」と同じ例ですが、マテリアルのスケールが全く違って見えるかと思えます。これはUVテクスチャマップの座標が異なる為です。

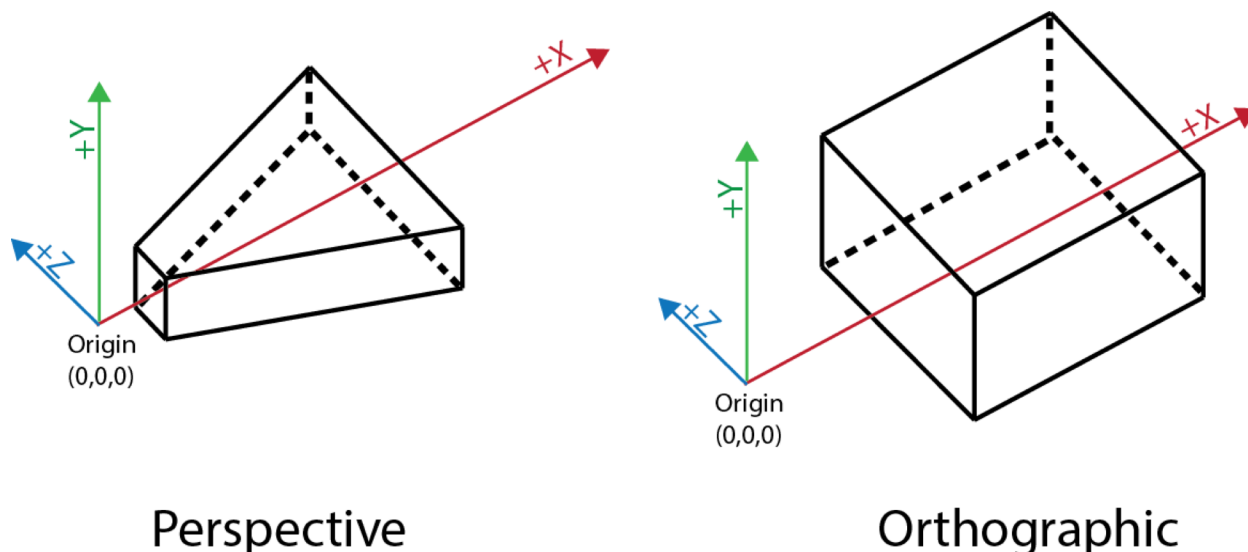
3Dシーンの二つ目の要素はライト（照明）です。実在の物と同様に、3Dシーンには光が必要です。サンプルファイル「Rendering_6.toe」では、一つの箱がレンダリングされているだけですが、Render TOPでは何も見ることはできません。これは、ライトのDimmer（鮮明度）が0に設定されているからです。これを見ればライトがどれだけ大切な要素かを理解して頂けるかと思えます。

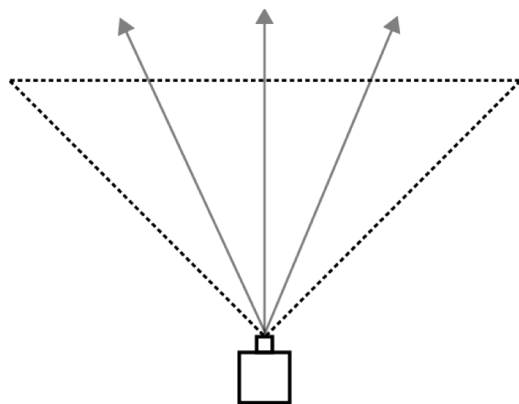
次の三つのサンプルファイルでは、先程と同じ例を、Light COMPの場所を変えて異なった角度からライトを当てています。「Rendering_7.toe」、「Rendering_8.toe」そして「Rendering_9.toe」で参照して下さい。

3Dシーンの最後の要素はカメラです。カメラがプロジェクトの目であり、視線であります。カメラに映ったもののそのものがレンダリングされるのです。サンプルファイル

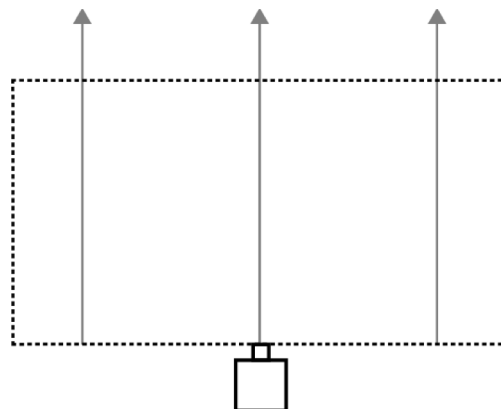
「Rendering_10.toe」を開いてください。このシーンの動きは全て、カメラの動きから来ています。一度カメラを設定してしまつて、その存在を忘れてしまい、3Dシーンが動きも少なく活気のないものになってしまうことは良くあります。映画カメラマンのつもりで、様々なカメラポジション、焦点距離、カメラの動き等を試してみましょう。

カメラの種類は主に二つ、Perspective（パースペクティブ・透視投影）カメラとOrthographic（オーソグラフィック・平行投影）カメラがあります。





Perspective



Orthographic

パースペクティブカメラは人間の目に似た働きをするので、コンセプトを理解するのはそれほど難しくないはずです。パースペクティブカメラには「視点」と「フラスタム（視野角）」が存在し、カメラが何を取り込み、そしてどの様にレンダリングするかが決まります。3Dシーンの光が全て視点に集中する、と考えると良いでしょう。パースペクティブ・コレクション（透視補正）で遠近を適用してオブジェクトが映し出されます。どういう意味かという、遠くにあるもの程小さく映し出されるのです。人間の目と同様に機能するので、このタイプの方が良く使われるカメラです。

「Camera_1.toe」はパースペクティブカメラの例で、パースペクティブ・コレクションに注目しています。二つの同じ大きさの立方体がありますが、実際に目で見ると同様に、遠くに位置付けられた立方体の方が小さく映し出されています。

一方、オーソグラフィックカメラは全く異なるもので、キャド（CAD: Computer-Aided Design）ユーザーには経験があるものかもしれません。このタイプのカメラの大きな特徴は、「視点が存在しない」という事です。パースペクティブカメラの様に、3Dシーンの光は一つの場所に集められることはなく、Z軸上の位置によってオブジェクトが歪むこともありません。簡単に言うと、カメラからの距離によってオブジェクトの大きさが変わることがない、という事です。

「Camera_2.toe」のサンプルファイルを開いて見て下さい。これは「Camera_1.toe」と同じ例ですが、オブジェクトの見え方に大きな違いがあります。この例では、Z軸上の位置には関係なく二つの立方体が隣に並んでいる様に見えます。このコンセプトを理解するのには時間がかかりますが、下記の二通りの考え方が助けになるかと思います。

一つ目。三次元の世界が、レンダリング前に平面化されること。

初代のスーパーマリオが実はオーソグラフィックカメラを使ってレンダリングされていた三次元シーンだったと考えてみて下さい。例えば、悪者は画面上どこに位置しようとは関係なく、画面中央にいるもの、画面端にいるもの、全て同じサイズと形で遠近は適用されず表示されていました。

二つ目。建築CADプログラムのブループリントを思い浮かべて下さい。これは正に3Dオブジェクトをそのまま平面上に表した物です。例えば、どこに視点があったとしても、同じモデルのドアは画面端にあっても、画面中央にあっても関係なく必ず同じサイズで表示されます。例えばカメラの位置が変わったとしてもレンダリングされるオブジェクトが歪むことはありません。

ところが、Z座標がオーソグラフィックカメラに全く無関係という訳ではありません。Z座標は他のジオメトリをレイヤーにして重ねる際に重要になります。

カメラはプロジェクションマッピングにおいても重要な存在です。後程プロジェクションマッピングについては更に詳しく説明しますが、ここではカメラの役目に注目しましょう。プロジェクションマッピングのメインゴールは、実在の物体を立体的なプロジェクションスクリーンとして利用することです。その為には、スクリーンとして利用する立体物（建物等）の3DモデルをTouchDesignerに取り込み、テクスチャを追加した後、レンダリングする必要があります。Camera COMPは実際のプロジェクターをシミュレーションすることができます。プロジェクターとレンズのスペックの詳細を入力することによって、TouchDesignerはプロジェクターのビューを正確に反映することができるのです。そのビューをレンダリングして出力し、そして実際のスクリーンとして利用する立体物と照らし合わせ、位置・サイズの調節が行われます。これがプロジェクションマッピングの基本となります。

7 COMPs

7.1 はじめに

COMP（コンポーネントオペレーター）には三つのカテゴリーがあります。

- 3D Object COMP は3Dシーンを作成し、ライトを加え、そして表示します。
- Panels COMP はユーザーインターフェイス（ボタン、スライダー、ウィンドウペイン等）を作成します。
- Other COMP は、それぞれ特殊な機能を備えており、キーフレームアニメーションを作成するもの、他のオペレーターを複製するもの、出力専用ウィンドウを作成するもの等があります。

通常、コンポーネントオペレーターは他のオペレーターと合わせて使用します。3D Object COMPはSOPや3Dシーンをレンダリングするのに様々な組み合わせで使用されます。Panels COMPはユーザーインターフェイスと出力ウィンドウを作成する為に使用されます。Other COMPはキーフレームアニメーション、オペレーターの動的レプリケーション、複数のディスプレイでウィンドウを開く等、様々なタスクをこなします。

そして更に興味深いのは、実はTouchDesigner自体の大半が、TouchDesigner内のコンポーネントで構成されている、という事です。これを理解できると、TouchDesignerをより深く知る手助けになるでしょう。例えば、全てのパネルコンポーネントは他のオペレーターで構成されています。Button COMPを作成してみると、そのネットワークの背景はText TOPでできています。そして、Panel CHOP がButton COMPのオン・オフ値を表示します。同様に、TouchDesignerのユーザーインターフェイスの全てはrootレベルで作成され、そして「ui」コンテナで保存されています。MIDI Mapper DialogやWindow Placement Dialogといったメニューアイテムやダイアログでさえも他のTouchDesignerのコンポーネントによって作成されています。

7.2 Window COMP

Window COMPはオペレーターのコンテンツを別ウィンドウで表示するもので、ほとんどのプロジェクトで必要不可欠です。Window COMPを使ってフルスクリーン表示をする場合でも、通常のウィンドウ表示をする場合でも、状況に応じて調整する必要があるパラメータオプションは数多くあります。一つ一つのプロジェクトは異なったユニークなものなので、全てのプロジェクトに合う一つの設定は存在しません。TouchDesignerのヘルプ（Wiki）でパラメータオプションを紹介していますので、それぞれの状況に最適な対応をできるように、時間をかけて研究することをお勧めします。

サンプルファイル「Window.toe」を開いて下さい。この例ではWindow COMPの基本機能と便利な使い方を紹介します。まず、Container COMPをWindow COMPの出力ソースとして使用するのが最適な方法だと言えます。何故なら、例えばパフォームモードでも、ビューア内でテキストチャをドラッグして動かすことができるからです。この場合、テキストチャを手動で元の位置に戻す、もしくは再び読み込みを行うまでビューア内のテキストチャは動いたままとなります。Container COMPではテキストチャをドラッグして動かすことはできません。

次は、「Change monitor」ボタンのシンプルな機能を紹介しましょう。「Change monitor」ボタンが押される度に、Window COMPのコンテンツを映し出しているモニターが、一つずつ順番に表示されます。これは、Monitor DAT内のモニター数をCount CHOPでループさせることで可能になったものです。

7.3 ユーザーインターフェイスコンポーネント

コンポーネントオペレーターはユーザーインターフェイスを作成するので非常に重要なオペレーターです。そして明確には、この役目を果たすのはPanels COMPです。後のチャプターで数々のユーザーインターフェイスの作成法が紹介されますので、ここではいくつか基本的なものを紹介したいと思います。

最も実用的なPanels COMPは次の三つです。

- Slider COMP
- Button COMP
- Container COMP

Slider COMPとButton COMPの機能は他のアプリケーションのスライダーやボタンと似たものですが、状況に応じてカスタマイズができます。Button COMPはトグルボタン、ラジオボタン、そしてモーメンタリーボタンとして設定する事ができます。Slider COMPは通常の一軸のスライダー、もしくは二軸のパッド形式のスライダー（カラーピッカー等）として利用できます。

対照的に、Container COMPは他のユーザーインターフェイスを保管するコンテナの役割以外には特に機能はありません。

「UI.toe」のサンプルファイルを開いて下さい。まず、シンプルなユーザーインターフェイス（container3）が見て取れます。そして、ネットワーク下部には二つのButton COMPと五つのSlider COMPがありますが、これらがユーザーインターフェイスの機能を作成しているコンポーネントです。これらの親であるContainer COMPはボタンとスライダーをグループにまとめて整理する為に使われています。「container1」と「container2」のビューアをオンにすると、ボタンとスライダーをその場で操作する事ができます。Container COMPが出力データやオペレーターをネットワークに持つことはありません。「container1」と「container2」が「container3」にま

とめられたとしても同じ結果です。これは、Container COMPが子供のオペレーターを自身のビューアに表示する機能を持っているからです。この様に、Container COMPは数々の小さなアイテムを合わせて複雑なインターフェイスを作成することができます。

8 MATs

8.1 はじめに

MAT（マテリアルオペレーター）は3Dジオメトリのマテリアルやシェーダーの為のオペレーターです。

マテリアルやシェーダーを扱う際にはコンピュータグラフィックやレンダリングについての深い知識が必要になります。この知識なしでは、TouchDesignerでの作業はPhong MATやPoint Sprite MATの基本的な設定だけに限られてしまうでしょう。

ジオメトリにテクスチャやUVマッピングを適用するのはとても複雑な作業なので、専門の3Dモデリングソフトウェアを使用する事で最良の結果が得られるはずです。テクスチャを付けたモデルをインポートすれば、TouchDesignerで使用する事ができます。

Texture SOP等、テクスチャを扱うツールがTouchDesignerには数々ありますが、専門のソフトウェアを使うことで複雑な作業に費やす時間は軽減されるでしょう。

8.2 Phong、GLSL、Point Sprite マテリアル

代表的なMATは次の三つです。

- Phong MAT
- GLSL MAT
- Point Sprite MAT

このチャプターではこの三つのMATの基本的な使い方を紹介します。この三つは全て、マテリアルとシェーディングの作業を行うものですが、それぞれが異なった機能を持っています。

Phong MATは最もよく利用されるマテリアルオペレーターで、3Dジオメトリにテクスチャを適用する為に使用します。Color、Bump、Specular、Diffuse等、多様な種類があります。Ambient、Diffuse、Specular、Emit、そしてConstant のライト（照明）の設定を自由に組み合わせることができます。「Phong.toe」のサンプルファイルを開いて下さい。Phong MATの簡単なサンプルを二つ紹介します。一つ目はテクスチャのアルファを使って透明のボックスを作成しています。二つ目はライトの状態には関係なくオブジェクトを完全に映し出すようにEmit Lightが1・1・1に設定されています。

GLSL MATは、OpenGLシェーディング言語（GLSL）を使って独自のマテリアルを作り出す為に使用されます。GLSLはとても優れたプログラミング言語で、複雑で計算スピードの速いテクスチャを作成する事ができます。これは、アセンブリ言語を使わずに、グラフィックスパイプラ

インに操作性を与えることで可能になります。使い始めは習得するのに時間がかかるかもしれませんが、GLSLシェーダーはインターネット上で沢山紹介されていますし、TouchDesignerフォーラムの「Shared Component」のページでも数々の実例が紹介されています。

Point Sprite MATはスプライトをパーティクルシステムのポイントに割り当てます。ポイント・スプライトとは、その名の通り、2D画像（スプライト）を3Dイメージの点（ポイント）に貼り付けるものです。スプライトは常にカメラを正面としており、Z座標の位置によってサイズが決まります。サンプルファイル「Point_Sprite.toe」を使って説明しましょう。例えば、これと同様のネットワークをポイント・スプライトなしに作成するとしたら、数え切れない程のTransform TOPとComposite TOPでごちゃごちゃしたネットワークになる上に、大容量のメモリが必要になります。パーティクルシステムとポイント・スプライトを使うことによって、ネットワークはまとまり良く、理解しやすくなり、メモリを多く使用することもあります。

8.3 UVマップ

高度な3DジオメトリにはUVマッピングは欠かせません。他の3Dモデルの要素と同様に、UVマップの作成と編集には専用のモデリングソフトウェアを使うのが最適です。

3Dジオメトリに静止画のテクスチャや動きのあるテクスチャを与える事ができるのがUVマッピングです。静止画とモーションが作成された二次元の世界と、ジオメトリの三次元の世界を繋ぐ役割を果たします。

UVマッピングには三つのステップがあります。一つ目は三次元のオブジェクトを切り開いて二次元の展開図にすることです。この平面化されたテクスチャがUVマップです。マップと呼ばれるのは、正に実際の地図の様に三次元のオブジェクトを正確に比例して展開した物だからです。これは通常の市街地図や世界地図も同様です。

二つ目のステップはテクスチャの編集です。デザイナーやアーティスト達は、専用のソフトウェアを使い、先程の二次元UVマップに静止画もしくはモーションのテクスチャを追加します。UVマップの利点は、高い精密度でのテクスチャの貼り付けが可能だという点です。

三つ目のステップは3Dジオメトリへのテクスチャの適用です。使用するソフトウェアによって、ここでの手順は様々です。

これらの三つのステップがUVマッピングです。

三つ目の「テクスチャの適用」はTouchDesignerでは比較的頻度の高い作業だと言えます。3Dジオメトリが専用ソフトウェアから正しくエクスポートされてさえいれば、UVマップを適用する場所を示した座標が含まれているはずです。その場合、テクスチャがMove In TOPへと読み込ま

れ、Phong MATによって3Dジオメトリへと適用されます。もしUVマップの適用に変更が必要な場合はTexture SOPを使います。

下の図はシンプルな3DボックスとそのUVマップの例です。

