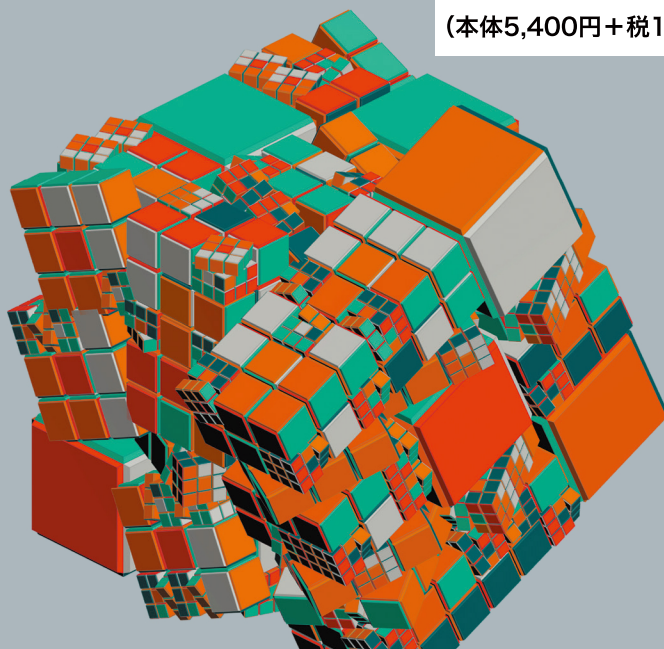


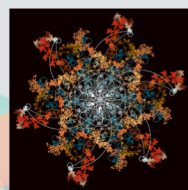
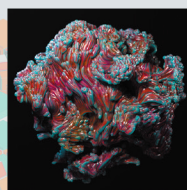
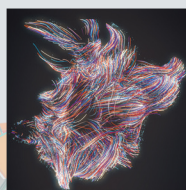
エクスプレッション さつき先生と学ぶはじめての



定価：5,940円
(本体5,400円+税10%)



ダウンロードデータ付属 ↓



初心者でもわかりやすいと評判の

Houdi
どんど

試し読み

for preview only

高瀬
紗月
著

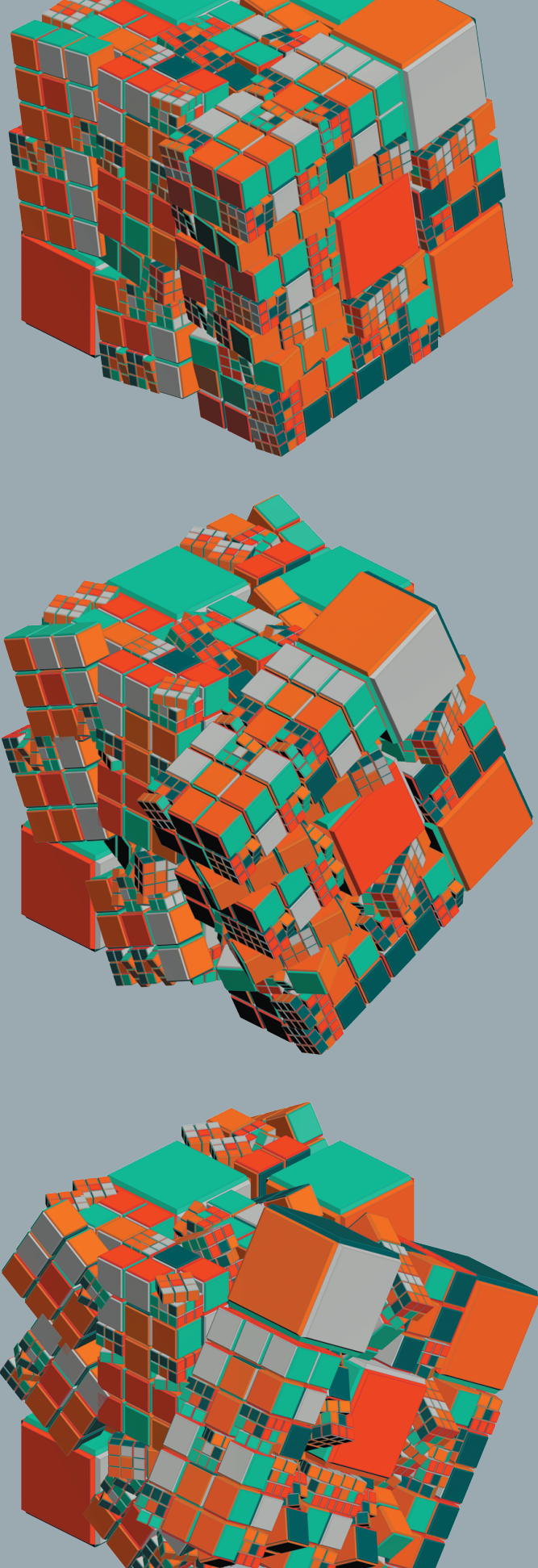


 SideFX[®] 公式

エロドリ

さつき先生と学ぶはじめての

高瀬 紗月 著



ダウンロードデータと書籍情報について

本書のウェブページでは、ダウンロードデータ、追加・更新情報、発売日以降に判明した誤植(正誤)などを掲載しています。

また本書に関するお問い合わせの際は、事前に下記ページをご確認ください。

<https://www.borndigital.co.jp/book/9784862465962/>

※ご使用している環境(OS)やアプリケーションのバージョンにより、一部画面キャプチャやUI名称、機能等が異なる場合がありますので、あらかじめご了承ください。

本書の対象読者

本書は、3DCGやHoudiniに興味があれば、どなたでも楽しめるものと考えています。特に対象としているのは、3DCGを学ぶ専門学生と、よりテクニカルな3DCGを学びたいと考えている現場の3DCGアーティストです。

そのため、基本的なコンピュータの操作と基礎的な高校数学を除き、前提知識を必要としません。Houdiniには、独特なユーザーインターフェースや独自のプログラミング言語などが存在しますが、それらもすべて必要になったタイミングで解説を入れていきます。

本書の読み進め方

後の章で、その前に完成させた作例を利用・発展・比較することがあるので、前から順番に読み進めてください。また、各章で扱う作例はサンプルファイルを用意していますので、パラメータ設定の詳細などは、そちらからもご確認いただけます。

● 本書に登場する様々な画像について

本書の手順をそのまま再現できれば、基本的に同じものが完成しますが、一部の章扉の画像などは、書籍用により高解像度に調整していたり、外部のソフトウェアを用いて簡単な色調補正や背景の合成処理を行うなどの都合上、同じものにならないものもあります。

筆者が調整した作例は、可能なかぎりサンプルファイルとして配布していますので、そちらをご確認ください。また、変更・調整してよいパラメータについては本文でその旨を記載していますので、ぜひみなさん自身で自由に変更して、自分だけの作品を作ってみましょう。

● 本書の動作環境

本書の解説は、以下の環境で行っています。

Houdini 20.5

Windows 11

■著作権に関するご注意

本書は著作権上の保護を受けています。論評目的の抜粋や引用を除いて、著作権者および出版社の承諾なしに複製することはできません。本書やその一部の複製作成は個人使用目的以外のいかなる理由であれ、著作権法違反になります。

■責任と保証の制限

本書の著者、編集者および出版社は、本書を作成するにあたり最大限の努力をしました。但し、本書の内容に関して明示、非明示に関わらず、いかなる保証も致しません。本書の内容、それによって得られた成果の利用に関して、または、その結果として生じた偶発的、間接的損傷に関して一切の責任を負いません。

■商標

本書に記載されている製品名、会社名は、それぞれ各社の商標または登録商標です。本書では、商標を所有する会社や組織の一覧を明示すること、または商標名を記載するたびに商標記号を挿入することは特別な場合を除き行っていません。本書は、商標名を編集上の目的だけで使用しています。商標所有者の利益は厳守されており、商標の権利を侵害する意図は全くありません。

はじめに

本書をお手に取ってくださり、ありがとうございます。本書は、Houdiniを使って3DCGの根幹から高度な自動処理まで、一歩ずつ段階を踏んで学ぶ書籍です。

みなさんは、「3DCG」という言葉にどのような印象をお持ちでしょうか？ 本物と見紛うようなハリウッド映画の3DCGを想像した方もいれば、3DCGゲームの仮想世界やAR(拡張現実)などを想像した方もいるかもしれません。

近年、コンピュータとそれに関連する技術の急速な進歩によって、3DCGは身近な技術になってきています。しかしながら、本格的な3DCGの制作となると、まだまだハードルが高い部分も少なくありません。見た目にもきれいな作品を作るとなれば、さらにハードルは上がります。

また、ソフトウェア(アプリ)では、数学や物理学などにおける専門的な言葉を使って説明されるものも多く、みなさまの中には一度挫折してしまったという方もいるでしょう。特にHoudiniは、「難しい」「一部の人が扱えるツール」などと思われがちです。

しかし、筆者はそうは思っていません。ただほかのソフトウェアと「違う」のだと思っています。Houdiniは極めて基本的な操作がオープンになっているので、その基本を知らなければ、なんの話をしているのかさっぱり理解できないのです。つまり、簡単とまでは言いませんが、一歩ずつ基礎から積み上げていけば、入門のハードルはさして高くありません。

そこで本書では、はじめに3DCGの基本的な工程を体験した後、その根幹にある概念を掘り下げ、最終的には複雑なアニメーションを大量に自動生成する仕組みを構築します。この独自の仕組みを構築することは、ポイント1個単位の細かい情報の一つひとつを管理できなければ叶わないことなので、Houdiniを通して3DCGの基礎の基礎を理解することの大きな意義を感じられるはずです。

本書によって、より根源的な3DCGの面白さが一人でも多くの方に伝わればと願います。また、すでにある程度3DCGを知っている方に対しては、その可能性を押し広げる一助となれば幸いです。

2024年8月 高瀬 紗月

著者プロフィール

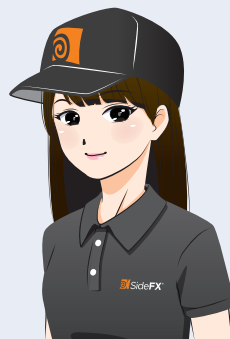
高瀬 紗月 (たかせ さつき)

SideFXのHoudini先生。高校時代Houdiniに出会い、以来様々な計算を作品へと応用することを模索。日本語でのHoudiniの解説動画を公開したことがきっかけで、SideFXのHoudini先生となる。

エフェクト・テクニカルアーティストの経験を活かし、Houdiniを導入する日本中の企業・学校に技術サポートやトレーニングを行う。一般向けの講演等でもわかりやすさとその技術力が認められ、「さつき先生」の愛称で親しまれている。計算でカッコいいものを作るのが得意。

X : @Satsuki_8198

YouTube : <https://www.youtube.com/@StudioSatsuki>



目次

はじめに	3
------	---

CHAPTER

01

はじめての3DCGを制作しよう

12

1-1 3DCGができるまで	14
1-1-1 モデリング(形作り)	15
1-1-2 マテリアル設定、テクスチャ制作(質感作り)	15
1-1-3 レイアウト(配置)	16
1-1-4 アニメーション、シミュレーション(動き作り)	17
1-1-5 ライト、カメラ設定	17
1-1-6 レンダリング(完成画像の計算)	18
1-2 Houdiniを使おう	19
1-2-1 ダウンロードとインストール	19
1-2-2 ユーザーインターフェース(UI)	23
1-2-3 Houdiniの基本的な考え方	24
COLUMN コンテキストの通称名	27
1-3 3DCGを作ろう	28
1-3-1 はじめに：こまめに保存しよう	28
1-3-2 モデリング	29
COLUMN 素早くツールを切り替えよう、実行しよう	33
COLUMN 変な操作をしてしまったら	34
1-3-3 レイアウト	35
1-3-4 マテリアル設定	37
COLUMN MaterialX ってなに？	37
1-3-5 ライト、カメラ設定	39
1-3-6 レンダリング	43
COLUMN マテリアルがしていたこと	45
COLUMN Diffuse / Reflection Limit はなにが違うの？	49
1-3-7 完成！	50
さつき先生小噺 大規模制作で、作業の分担や管理はどうするの？	51

2-1	プロシージャルな考え方	54
2-1-1	単純な「手続き」から見てみよう	54
2-1-2	よりよい「手続き」を考えよう	57
2-1-3	押し出す面の指定方法を見直そう	58
2-2	ジオメトリの構造を理解しよう	59
2-2-1	ジオメトリを構成しよう	59
2-2-2	Houdiniで確認しよう	60
2-3	アトリビュートを理解しよう	64
2-3-1	アトリビュートを確認しやすくしよう	65
2-3-2	アトリビュートを使おう	65
COLUMN	原始的な操作をもう少し深掘り	70
2-3-3	アトリビュートの使い道を知ろう	70
COLUMN	アトリビュートのClassを視覚的に確認しよう	72
さつき先生小喃	問題解決! 浮いてしまうボール?	76

3-1	実装方法を考えよう	82
3-1-1	必要な操作をまとめよう	83
3-1-2	実装方針をまとめよう	84
3-2	Houdiniで実装しよう	85
3-2-1	原点にポイントを追加しよう	85
3-2-2	原点周りを回転するポイントを追加しよう	86
COLUMN	右手座標系と左手座標系	88
3-2-3	追加したポイントを移動しよう	89
3-2-4	ループさせよう	91
COLUMN	様々な「繰り返し」	100
COLUMN	相対参照と絶対参照	104

3-2-5 線を追加しよう	105
3-2-6 様々なパターンを生成しよう	106
3-2-7 アートとして完成させるには	107
さつき先生小囁 コンピュータの計算能力と高速化	112

CHAPTER

04 シミュレーションってなに？

114

4-1 シミュレーションについて考えよう	116
4-1-1 ボールの軌道をシミュレーションしよう	116
4-1-2 シミュレーションの意味を理解しよう	117
4-2 未来の予測は簡単ではない	118
4-2-1 少し先の未来を考えよう	118
4-2-2 フィードバックループとシミュレーション	120
4-3 Solverノードを使ってみよう	121
4-3-1 Solverノードを追加してみよう	121
COLUMN 最初のフレーム、どうするか問題	123
4-3-2 「初期状態」を準備しよう	124
4-3-3 次の状態を計算しよう	124
4-3-4 結果を確認しよう	125
4-3-5 シミュレーションと Solverノードについてのまとめ	126
4-4 Solverノードで面白い動きを作ろう	127
4-4-1 今回作るもの	127
4-4-2 ステップ1：初期状態の準備をしよう	127
4-4-3 ステップ2：次の状態を計算する処理を作ろう	130
COLUMN パラメータが多すぎて見つからない！	132
4-4-4 ステップ3：シミュレーションを調整しよう	137
4-4-5 おまけ：アニメーションとして保存しよう	139
さつき先生小囁 大規模なシミュレーションに必要なマシン性能は？	143

5-1	サンゴ風の成長アニメーションを作ろう	148
5-1-1	成長を表現するアルゴリズム	148
5-1-2	成長を表現するアルゴリズムのまとめ	149
5-2	Houdiniで実装しよう	150
5-2-1	最初の形状を作成しよう	150
5-2-2	曲率を計算しよう	158
5-2-3	ポイントを法線方向に移動しよう	158
5-2-4	処理を繰り返そう	160
5-2-5	エラーを修正しよう	161
5-2-6	処理が重い原因を考えよう	162
5-3	ボリュームとは	164
5-3-1	2次元から考えてみよう	164
5-3-2	3次元になったらどうなるの?	165
5-3-3	3次元形状を表現するボリューム(SDF)	165
COLUMN	勾配の定義と直感的なイメージ	167
5-3-4	ボリュームについてのまとめ	167
5-4	ボリュームで問題を解決しよう	169
5-4-1	ネットワークを整理しよう	169
5-4-2	ポリゴンメッシュをSDFボリュームに変換しよう	170
5-4-3	すべてをボリュームで処理しよう	173
5-4-4	ボリュームによる問題解決のまとめ	175
5-5	ボリュームで実装し直そう	175
5-5-1	前処理をしよう	176
5-5-2	メインの処理を実装しよう	176
5-5-3	ここまでの結果を確認しよう	182
5-5-4	曲率で成長量をコントロールしよう	183

5-6 アーティスティックなコントロール	188
5-6-1 一樣すぎる成長を改善しよう	188
COLUMN 別の形をもとに成長を抑制するには?	190
5-6-2 空間全体での「流れ」を作ろう	190
5-6-3 小さな改良と修正をしよう	201
5-6-4 今後の発展とまとめ	215
さつき先生小噺 執筆がやばいの話	222
さつき先生小噺 いいカメラのすすめ	223

CHAPTER

06

Solarisを使ってみよう

224

6-1 Solarisについて知ろう	226
6-1-1 Solarisとは	226
6-1-2 レンダリング方法は2種類ある	227
6-2 Solarisに「はじめてのシーン」を持てこよう	227
6-2-1 Scene Import ノード	228
COLUMN Karma CPUとKarma XPUはなにが違うの?	230
6-2-2 少しだけUSDを理解しよう	234
6-2-3 先ほどはなにが問題だったのか	235
6-2-4 レンダリングしよう	237
6-2-5 USDファイルを保存しよう	240
COLUMN outにあったKarmaもLOPで動いている	242
6-3 Solarisでシーンを構築しよう	243
6-3-1 ジオメトリだけを読み込もう	243
6-3-2 マテリアルを付けよう	245
6-3-3 ライトを置こう	249
6-3-4 カメラを置こう	254
6-3-5 Render Galleryを活用しよう	256
6-3-6 マテリアルの調整前に	258
6-3-7 マテリアルを調整しよう	260
6-3-8 レンダリングしよう	272
さつき先生小噺 CGI or NOT	279

7-1	Attribute Wrangle ノードを使おう	284
7-1-1	まずは1行書いてみよう	284
COLUMN	コードの拡大縮小	286
7-1-2	Run Overを変更しよう	287
7-2	基本的な概念を学ぼう	289
7-2-1	変数：番地の名付け	290
7-2-2	関数：便利な処理をまとめたもの	294
7-2-3	if / else 文：条件分岐	297
7-2-4	while ループ：繰り返し処理させよう	299
7-2-5	for ループ：少し便利に使えるループ	300
COLUMN	様々なループの使用例	301
7-3	ジオメトリを構成しよう	302
7-3-1	ポイントを追加しよう	303
7-3-2	プリミティブの追加と頂点を登録しよう	304
7-4	VEX で書き直そう	305
7-4-1	処理を復習しよう	305
7-4-2	必要な変数を用意しよう	307
7-4-3	ポイントを追加しよう	308
7-4-4	プリミティブを追加しよう	310
7-4-5	アトリビュートの設定と色を着けよう	311
COLUMN	「Curve U Attribute」のU	312
7-4-6	変数をHoudiniのパラメータにしよう	314
7-5	以前の実装と比べてみよう	317
7-5-1	小さな違いを確認しよう	317
7-5-2	見た目が変わった!?	318
さつき先生小噺	PythonとOpenCL C 言語	323

CHAPTER

08

立体回転パズルを作ろう

324

8-1	実装前の整理をしよう	326
8-1-1	用意するパラメータ	326
8-1-2	実装方針を考えよう	328
8-2	モデリングをしよう	328
8-2-1	下準備をしよう	328
8-2-2	サブキューブを作成しよう	329
8-3	初期状態を用意しよう	338
8-3-1	状態の表現方法を理解しよう	338
8-3-2	同じvector型でも違いがある？	342
8-3-3	回転記号の文字列を加工しよう	344
8-4	回転の処理の仕組みを作ろう	351
8-4-1	Subnetwork ノードに処理をまとめよう	351
8-4-2	処理する情報を取得しよう	353
8-4-3	例外処理をしよう	354
8-4-4	回転を設定しよう	356
8-4-5	各回転後における状態を事前計算しよう	362
8-5	アニメーション情報を計算しよう	370
8-5-1	アニメーションの生成方法	370
8-5-2	現在処理すべき状態id	372
8-5-3	回転動作の完了割合	375
8-5-4	例外処理について学ぼう	377
8-5-5	アニメーション開始のタイミング	378
8-5-6	パラメータの追加	379
8-5-7	パラメータを整理しよう	382
8-6	コントロールと書き出し	383
8-6-1	アーティスティックなコントロール	383
8-6-2	プレビューを書き出そう	397
	さつき先生小囁 HDA	404

9-1	TOP ? PDG ?	408
9-1-1	やるべきこと≡パラメータ情報	409
9-1-2	TOPが管理すること／行うこと	409
9-2	TOP / PDGの操作に慣れよう	411
9-2-1	スケジューラーってなに？	411
9-2-2	ワークアイテムについて	413
9-2-3	処理を実行しよう	419
9-3	作業の前に	422
9-3-1	工程を確認しよう	423
9-3-2	TOPネットワークはどこにでも作れる	423
9-4	ワークアイテムを生成しよう	425
9-4-1	.csvファイルを読み込もう	425
9-4-2	階層化の切り替えアトリビュートを追加しよう	426
9-4-3	ワークアイテムアトリビュートを利用しよう	428
9-5	処理を実行しよう	430
9-5-1	ワークアイテムをフィルタリングしよう	430
9-5-2	OpenGLでレンダリングしよう	431
COLUMN	少しだけ中身を見てみよう	432
9-5-3	In-Processクッキングを使おう	443
9-5-4	COPによる合成処理をしよう	444
COLUMN	どうして画像の色が暗くなるの？	450
9-5-5	FFmpegで動画ファイルを生成しよう	467
9-5-6	すべての処理をまとめて実行しよう	471
さつき先生小噺	さつき先生ができるまで	474
	SideFXとHoudiniの歴史 (Kim Davidson)	480
	索引	488

CHAPTER

01

はじめての3DCGを制作しよう

一口に3DCGを制作すると言っても、そこには様々な工程が存在します。本章では、主な工程を整理して、実際に簡単な3DCGを制作してみます。

SECTION

1-1

3DCGができるまで →P.14

- 1-1-1 ■モデリング(形作り)
- 1-1-2 ■マテリアル設定、テクスチャ制作(質感作り)
- 1-1-3 ■レイアウト(配置)
- 1-1-4 ■アニメーション、シミュレーション(動き作り)
- 1-1-5 ■ライト、カメラ設定
- 1-1-6 ■レンダリング(完成画像の計算)

SECTION

1-2

Houdiniを使おう →P.19

- 1-2-1 ■ダウンロードとインストール
- 1-2-2 ■ユーザーインターフェース(UI)
- 1-2-3 ■Houdiniの基本的な考え方

SECTION

1-3

3DCGを作ろう →P.28

- 1-3-1 ■はじめに：こまめに保存しよう
- 1-3-2 ■モデリング
- 1-3-3 ■レイアウト
- 1-3-4 ■マテリアル設定
- 1-3-5 ■ライト、カメラ設定
- 1-3-6 ■レンダリング
- 1-3-7 ■完成！

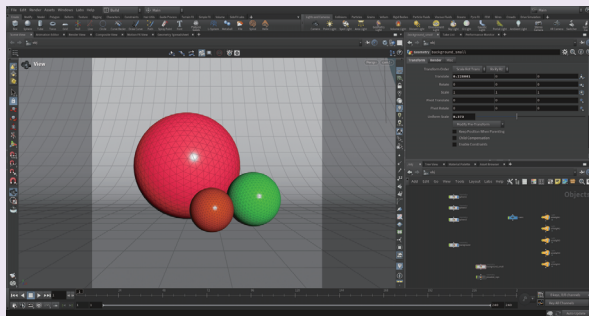


本章の作例です。このサンプルファイルは、ダウンロードデータの「01_TheFirstScene.hip」からご確認ください。



さつき先生

突然ですが、この画像は写真ではなくCGできています。本章では、このような画像を3DCGで作ることを目標に進めていきます。



ゆうか

ちょっとハードルが高くないですか……？



一歩ずつ丁寧に手順を追っていけば大丈夫です。そのために私もいるのです！

SECTION 1-1

3DCGができるまで

個人や制作会社によって、またプロジェクトによっても少しずつ異なりますが注1、主な工程は次の通りです。

- モデリング(形作り)
- マテリアル設定、テクスチャ制作(質感作り)
- レイアウト(配置)
- アニメーション、シミュレーション(動き作り)
- ライト、カメラ設定
- レンダリング(完成画像の計算)

ただし、必ずしもすべてがこの順番で行われるわけではありません。たとえば、質感作りは形状がないとできませんが、簡単なレイアウトやカメラワークの確認には、最終品質の形状やアニメーションは必要ありません。また、後になって前の工程を少し修正することもあります。

こういった基本的な工程を確認した後、先ほどの画像の制作工程をたどってみましょう。

注1 写実的な3DCGと2Dアニメ風CGの場合では、質感の設定が少し異なるなど、求めるテイストによって多少の差異がある。

1-1-1 モデリング(形作り)

まずは、**モデリング**という形を作る作業を行います。ポリゴンと呼ばれる多角形の面を張り合わせて目的の形を作る手法は、**ポリゴンモデリング**です。

ポリゴンモデリング以外にもモデリングの手法は多数あり、**スカルプティング**と呼ばれる手法では、「彫る」という意味通り彫刻をするように造形します。また、比較的新しい手法の**フォトグラメトリ**では、様々な角度から撮影された大量の写真を解析・統合して3Dモデルを作成します。



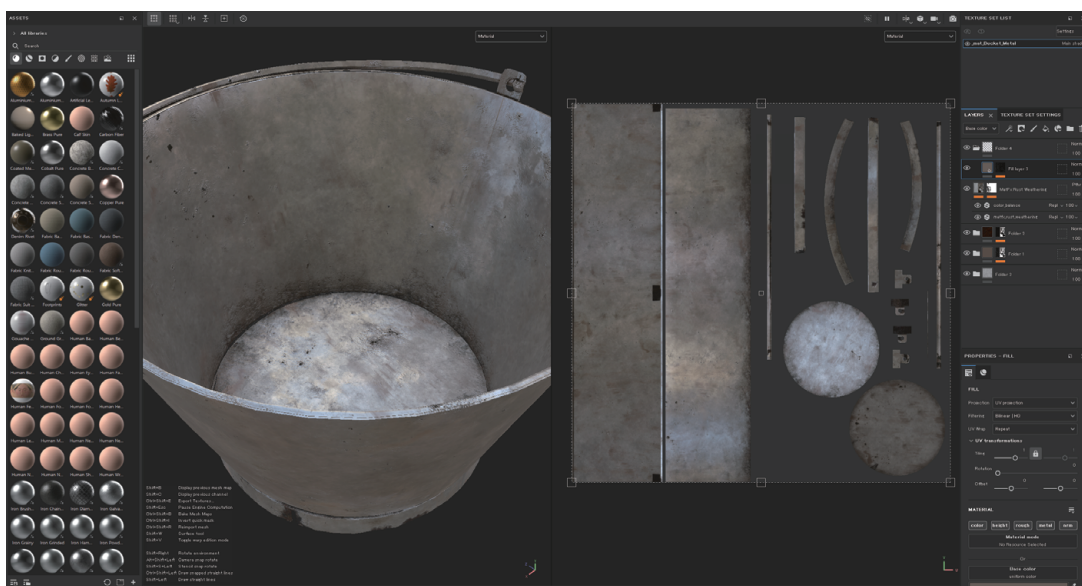
1-1-2 マテリアル設定、テクスチャ制作(質感作り)

現実世界には、プラスチック、ガラス、木材、金属など、様々な質感が存在します。これらを模倣するために、ほとんどの3DCGソフトウェアには質感を設定する項目があります。



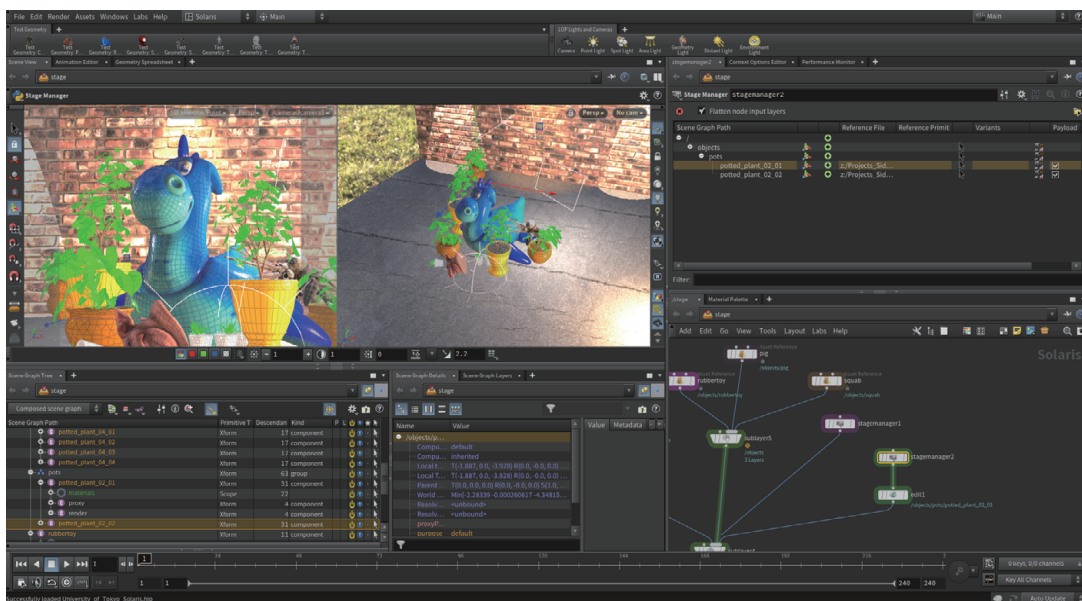
マテリアル：直訳で「材料、素材」といった意味があります。3DCGにおいては、ソフトウェア上で作られた質感の設定情報のことです。マテリアルは、一般的に多くの**シェーダ**から構成され、シェーダは、ある状態における色や光の**計算方法**を指します。

テクスチャ：直訳で「(物の)質感」といった意味があります。3DCGにおいては、マテリアルを設定する際に用いる画像ファイルのことで、小さなキズやそれに伴う細かな凹凸など、モデリングで表現するのがあまり現実的ではない部分などに使用します。



1-1-3 レイアウト(配置)

ある程度3Dモデルが作れたら、いよいよ配置していきます。ソフトウェアによって操作方法是異なりますが、作業自体は大体同じです。追加や修正をする場合は、前の工程に戻ります。



1-1-4 アニメーション、シミュレーション(動き作り)

3DCGに動きを付ける手法は多数あり、最も直感的に理解しやすいのは、**モーションキャプチャ**でしょう。体にセンサーを付けるなどして、実際の動きを3D空間上で再現します。手軽な手法に思えますが、実際に高い精度を得るにはそれなりに大規模な設備が必要だったり、エラーや迫力に欠ける部分を修正する必要があります。

現実に行うことが難しいキャラクターの動きは、**キーフレーム**という概念を用いて、アーティストが手動で設定して動きを付けることも多々あります。

現実に行うことが難しく、またアーティストが手動で設定することも難しい動きについては、**シミュレーション**が行われることもあります。これは、主にコンピュータの計算によって、爆発や水などの流体、服や毛の揺れ、魔法のような現実では起こりえないことのアニメーションを作る手法です。



※本章で作る作例は単純な静止画のシーンなので、今回これらの工程はスキップします。



ちなみに、シミュレーションでいろいろな動きを作るのが私の得意分野です。

1-1-5 ライト、カメラ設定

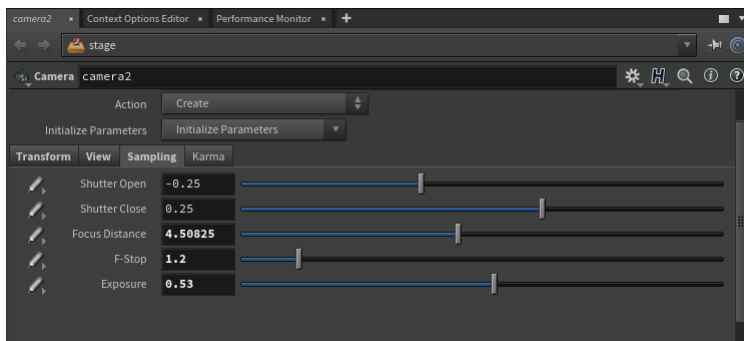
現実世界と同じように、ライトやカメラも設定する必要があります。ここで重要なのは、**現実的ではない設定もできる**という点です。

たとえば、通常昼間に屋外を撮影すれば、基本的に光源は太陽しかありませんが、天気や場所によって周辺環境が違ってくため、受ける光の色や強さも大きく異なります。しかし、カメラから見える範囲のものすべてを詳細にモデリングすることは現実的ではありません。

そこで3DCGのライトでは、単純な四角形や円形のライトだけでなく、**HDRI** (High Dynamic Range Images) と呼ばれる、光についてより多く、より広い範囲の情報を持った形式で全天球画像を撮影し、それを用いて環境による光の特性を近似します。



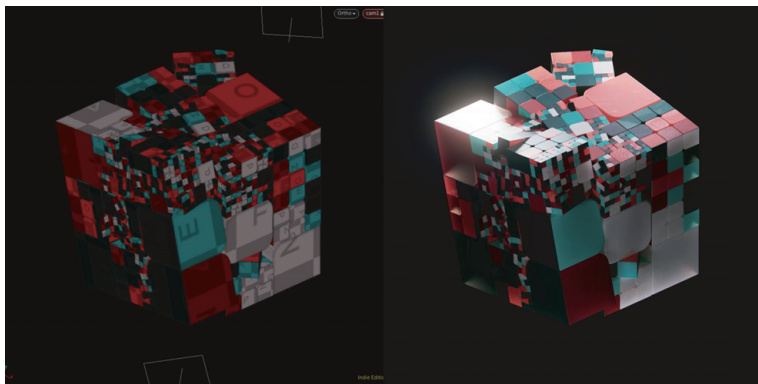
カメラにも、同じように様々な設定が可能です。現実世界のカメラにおける様々な設定値は、物理的な問題などから取りうる値に制約がありますが、3DCGにおいては、それらを多少アーティスティックにコントロールする目的で、現実ではあり得ない値に設定する場合があります。つまり、写実的な結果を得たい場合は、ある程度現実世界のカメラの知識が必要だということです。



1-1-6 レンダリング(完成画像の計算)

3Dモデルやその質感、ライトやカメラなどを含むシーンの情報をもとに、最終的な画像を計算して出力する工程を、**レンダリング**と呼びます。特に写実的な画像を作成したい場合、レンダリングは現実世界に飛び交う無数の光線をシミュレートすることに近い工程です。

レンダリングにかかる時間は、尺やシーン、レンダリングの手法、使用するマシンの性能などに大きく依存しますが、長いものだとな数時間～数日かかる場合もあります。レンダリングを行うソフトウェアを、**レンダラ**と呼びます。



Houdiniは、カナダのSide Effects Software, Inc. (通称SideFX) が開発する3DCGソフトウェアです。ハリウッド映画をはじめとする世界中の制作現場では、特にその柔軟性と生産性向上への貢献において、高く評価されています。

本書でHoudiniを扱う大きな理由は、ユーザーに対して3DCGにおける操作を非常にオープンに提供しているためです。後の作例では、非常に細かい話や、それに伴い基礎的な操作が必要な場面が多々ありますが、そういった要求に応えるという点において、Houdiniが非常に優れていると筆者は考えています。

1-2-1 | ダウンロードとインストール

ここでは、Houdiniのダウンロードおよびインストール方法について解説します。

ダウンロード：必要なファイルをインターネット上から手元のコンピュータにコピーして、保存する

インストール：保存したファイルを適切な場所に展開、コピーし、コンピュータに備え付けることで利用可能な状態にする

- 1 Houdiniには、Houdini Apprenticeという、学習用途に使える無料のバージョンがあります。SideFXの日本語公式サイト (<https://www.sidefx.com/ja/>) から[Get] > [ダウンロード]と進み、ダウンロードしましょう。

