HOUDINI FUNDAMENTALS

ノード、ネットワーク、および デジタルアセット

Houdiniのノードベースのワークフローを理解するための優れた方法は、あるプロジェクトの文脈でそのワークフロー を調べてみることです。プロシージャルな思考方法や操作方法を学び始めることが重要です。このレッスンでは独自 のカスタム**ブロック化**ツールの作成方法を学びますが、その際には、プロシージャルなノードやネットワークを使っ てツールの機能やインターフェースを定義します。

その過程で、Houdiniのワークスペースのさまざまな側面を利用することになります。必ず前書きの概要を参照し、 これらの UI 要素がどのように連携動作するのかを確認するようにしてください。レッスンはアイデアを実行に移す 良い機会を与えてくれますが、それこそが、学ぶための最良の方法の1つなのです。

レッスンの目標

• 任意の 3D 形状をおもちゃのブロックに変換するカスタムツールを作成すること

学習内容

- プラスチック製の連結可能なブロックをモデリングする方法
- デフォルトのゴム製おもちゃの形状からポイントのグリッドを 派生させる方法
- パックプリミティブとインスタンス化を使ってインタラクションを 高速化する方法
- テクスチャマップを使ってブロックにカラーを追加するために アトリビュートを使用する方法
- ノードやネットワークを操作してデータフローを 制御する方法
- ソリューションをパッケージ化して他のユーザと 共有するためにデジタルアセットを作成する方法
- ブロックが時間の経過に従って現れる アニメーションを作成する方法

レッスンの互換性

Houdini 17.5.293 以上に含まれる機能向けに 執筆しています	·
このレッスンの手順は以下の Hou 製品を使って実行できます。	d

Houdini Core	V
Houdini FX	ŕ
Houdini Indie	V
Houdini Apprentice	V
Houdini Education	•

ドキュメントバージョン 1.0 | 2019 年 7 月 © SideFX Software

パート 1: 単一のブロックを作成する

最初に、単一のブロックモデルを構築します。このブロックは後で、ブロック化された形状を作成する目的で、各ポイントに コピーされることになります。ここでは、いくつかのポリゴンモデリングツールを組み合わせてこの形状を作成します。 その過程で、アクションを実行するたびに対応するノードが Houdini 内に作成されていく様子を確認することができます。 これらのノードでは、ジオメトリの作成に必要な手順のレシピが作成されます。



このメニューはeditノードに関連付けられたものです。各ノードには、 対応するツールがアクティブである限りアクセス可能な独自のイン ターフェースが用意されています。この場合は、Move ツールによっ てハンドルへのアクセスが提供されています。他の場合には、Handle ツールを使ってノードのインタラクティブハンドルへのアクセスが提 供されます。



05 cを押して Radial メニューを表示させ、[Model]>[Polygons]> [PolyExtrude] を選択します。Scene ビューのペインで、 ハンドルを使ってポリゴンを上方にドラッグし、[Distance] を 0.05 に 設定します。

ボックスを裏返し、**s を押して**選択モードに移行した後、ボックスの 底面の 4 つのポリゴンを選択します。**q を押して PolyExtrude** ツール を繰り返します。パラメータペインで、スライダを使って [Inset] を 0.025 に設定します。

qを押してそのツールを繰り返し、[**Distance**]を-**0.175**に設定します。 完了したらまた裏返し、ブロックの上面が見える状態にします。

Rest Philar & Banaca Banahar & Company and Compan

3を押してエッジ選択モードに移行し、nを押してすべての エッジを選択します。Scene ビューで Tabキーを押し、 Group... と入力し始めます。[Group]を選択し、パラメータペインで [Group Name] を bevel_edges に設定します。次に、[Base Group] の [Enable] をオフに設定した後、[Include by Edges] セクションの [Enable] をオンに設定します。[Min Edge Angle] をオンにして 89 に設定した後、[Max Edge Angle] をオンにして 91 に設定します。



OB s を押して Select ツールに移動します。9 を押して [Select Groups] オプションをオンにします。ポップアップウィンド ウで bevel_edges グループをクリックします。

ビューポートで、**cを押して** Radial メニューを表示させます。この メニューから [Model] > [Polygons] > [PolyBevel] を選択します。 すると、polybevel ノードが追加され、[Group] フィールドに自動的に bevel_edges が設定されます。

次に、ベベルの [Offset] を 0.006 に設定します。[Fillet] の下で、 [Shape] を [Round] に、[Divisions] を 3 に設定します。



09 オブジェクトレベルに移動し、ネットワークビューでオブ ジェクトの名前を single_brick に変更します。ブロックが選択 された状態で Shift + を押し、この形状のサブディビジョンサーフェス 表示をオンにします。ブロックの選択を解除してモデルのサブディビ ジョン表示を確認します。オブジェクト上にワイヤ線が見える場合は v を押し、Radial メニューから [Shading] > [Smooth Shading] を選 択して非表示にします。

次のステップでは、ブロックのコピー先となるポイントを作成しま す。ここでは、ネットワークビューでノードの上にマウスを置き、 **表示**フラグを選択してブロックを非表示にしておきます。

作業内容を**保存** (**[Save]**) します。

サブディビジョン表示

0

Shift +/Shift - を使えば、選択されたポリゴンオブジェクトのサブ ディビジョン表示のオン/オフを切り替えることができます。これに よって作成されるビューポートのサブディバイドを使えば、形状の サブディビジョン表示がどのように見えるかを確認することができ ます。これらのホットキーは、オブジェクトレベル (オブジェクトの [Render] タブ)に存在する [Display As] パラメータを設定します。

同じタブ上の [Render Polygons As Subdivision Surfaces] をオン にしない限り、オブジェクトがサブディビジョンでレンダーされる ことはありません。



パート 1: 単一のブロックを作成する

パート 2: ポイントクラウドにブロックをコピーする

ここではまず、特定のジオメトリの形状に一致するポイントのクラウドを作成します。次に、3D グリッドに 対してブロックをインスタンス化することで、ブロック化バージョンを作成します。インスタンス化を実行する 際には、ブロックのジオメトリをパックした後で、各ポイントに対してブロックをインスタンス化します。

Scene ビューで Tab キーを押して **Test Geometry...** と入力 し始めた後、**[Test Geometry:Rubber Toy]** を選択します。 Series Societaries Societaries Societaries Spectrum Fort Societaries Links Rodal Ex Contractific Column Particle Senses Senior Reproduce Applications Senses Fueld Gauss Restitutes Paperson San Participations Paperson San Participation Senior Fuel Contractor Paperson San Participation Senior Fuel Contractor Paperson Participation Participation Senior Fuel Contractor Participation Participat Enter を押して原点の位置に配置します。 **スペース+b**を押して4つのビューを表示させます。スペース+hを使っ て正投影ビューをホーム設定し、ゴムのおもちゃ (ラバートイ) がすべ てのビューに表示されるようにします。 iを押してテストジオメトリオブジェクトに入り、以下の設定を行い ます。 [Uniform Scale]: 3 [Translate Y]: 1.5 これによってゴムのおもちゃの高さが高くなり、地面より上に表示さ れるようになります。再度スペース+h を使ってビューのホーム設定を 行います。 ネットワークエディタで、rubbertoy ノードの出力上で右マ Antonia and an and Antonia and Antonia **ウスボタンをクリック**し、Points from Volume... と入力し ます。[Points from Volumes] を選択し、そのノードをネットワーク 内に配置します。続いて、この新しいノードの出力に注目するため、 その表示フラグを設定します。 ビューの上にマウスを置き、スペース+bを押してこのビューを拡張し ます。sを押して選択ツールに移動し、2を押してポイント選択に移行 した後、nを押してすべてのポイントを選択します(黄色で強調表示さ れます)。これらのポイントが、ブロックのコピー先となります。 * 1 🖬 🖩 🖽 🔊 🖉 🛢 🔍 🖬 uを押してオブジェクトレベルに戻り、ゴムのおもちゃの名 前を短縮して rubbertoy にします。ネットワークビューで、 single_brick オブジェクトを表示させた後、Shift キーを押し、rubber toy をクリックしてから single_brick をクリックします。 [Modify] タブから [Combine] を選択して、これらのオブジェクトを マージします。ジオメトリレベルに移動し、これらのノードが merge ノードの入力に接続されます。 ネットワークビューで、*display merge* ノードを選択し、**Delete**



04 ネットワークビューで、display_mergeノードを選択し、Delete キーを押します。すると、polybevelノードのチェーンは表示 され、他方のチェーンは表示されなくなります。Houdiniでは、オブ ジェクトレベルに戻ったときに表示されるノードをどれにするかを選 択することができます。 The move is a second se



05 polybevel ノードの出力で右マウスボタンをクリックし、copy to points... と入力し始め、[Copy to Points] ノードを選択 します。クリックして2つのチェーンの下にノードを配置し、その表 示フラグを設定します。

[**Pack and Instance**] オプションをオンにします。このオプションは 重要です。これをオンにすると、コピーされたブロックの表示速度が オフ時に比べて大幅に向上するからです。

この時点ではノード上でエラーが発生しています。2番目の入力が接続 されていないからです。

pointsfromvolume ノードの下にあるドットをクリックし、これを copytopoints ノードの2番目の入力に接続します。

ブロックが一部重なり合っているように見えます。pointsfromvolumeノードに戻り、[Point Separation] を 0.2 に設定します。

これで、グリッドのポイントにブロックがコピーされるようになりま した。

O7 copytopoints ノードをクリックします。一部のブロックが暗灰色で表示され、適切に表示されない可能性があります。 その原因は、ビューポート内でのジオメトリカリングにあります。 これは表示設定で修正できます。Sceneビューで、スペース+dを押し て [Display Options] を表示させます。[Optimize] タブをクリックし た後、[Scene Polygon Limit]を50 millionに設定するか、[Distancebased Packed Geometry Culling] をオフに設定します。

これで、すべてのブロックがパックインスタンスとしてポイントにコ ピーされるようになります。

○8 作業内容を保存する前にネットワークを整理しておきましょう。単一ブロックを構成するノードを選択し、Shift+0を押してそれらのノードを囲むネットワークボックスを作成します。ボックスのタイトルバーをクリックし、single brick と入力します。そしてこのボックスを折りたたんで下方に移動すれば、ネットワークが若干すっきりとします。

ここまでの作業内容を**保存** ([Save]) します。

パックインスタンス

0

copy to points ノードの **[Pack and Instance]** 設定を**オフ**のままにした 場合、100 万個を超える数のポイントやプリミティブを含む大きなモデ ルが生成されます。この場合、インスタンス化が使用されないので、 ビューポート内での操作が非常に低速になります。

これを**オン**にした場合、ブロックモデル上の**338個のポイント**がパック されたうえでインスタンス化されるので、copy to points ノードから見 た場合のポイント数が非常に効率的なものとなります。

		1,024,	,816 C	enter.	Θ,	2.025,	Θ	
	Primitives	1,018,			-3.1,	-0.5,		
					3.1,	4.55,		
					6.2,	5.05,	5.4	
		_						
		Pack	and In	stan	ce	オフ		
		acit	and m	- cuii				
•			anam	Jean				
۲	Points	3,032	Center	0	, 2.02	25,	9	
¢	Points Primitives	3,032 3,032	Center Min	0 -3.1	, 2.02 , -0	25, -2.°	9 7	
¢	Points Primitives Vertices	3,032 3,032 3,032	Center Min Max	0 -3.1 3.1	, 2.02 , -0 , 4.5	25, -2.° 55, -2.°	0 7 7	

Pack and Instance オン





パート3 カラーを追加する

ここでは、ポイントにカラーを追加します。このカラーはインスタンス化されたブロックに引き継がれます。 最初、このカラーのトランスファーはビューポート内でしか適用されませんが、[Use Packed Color] オプションを含む 適切なマテリアルをセットアップすることで、ブロックをレンダーするためのポイントカラーをセットアップできます。



pointsfromvolumeノードとcopytopointsノードの間にcolorノー ドを追加します。すると、ポイントにカラーが追加され、そ のカラーがブロックにコピーされます。[Color] を赤色に変更し、背景 に対してブロックが目立つようにします。

Render Region ツールをクリックし、ゴムのおもちゃの上でバウンディ ングボックスをドラッグします。すると、テストレンダーが起動され ますが、ご覧のようにブロックは赤色でレンダーされません。カラー をピックアップするためのマテリアルを割り当てる必要があります。



ネットワークビューで、Tab キーを押して Material Network... と入力します。[Material Network] ツールを選択し、クリッ クしてネットワーク内にノードを配置します。

Material Paletteに移動し、/mat を閉じて /matnet を開きます。Principled Shader を matnet 内にドラッグします。それに brick_material という名前 を付けます。



ネットワークビューの戻るボタンを使ってジオメトリネッ トワークに戻ります。copytopoints ノードの出力で右マウス ボタンをクリックし、Material... と入力します。[material] を選択 し、そのノードをネットワーク内に配置し、その表示フラグを設定し ます。

[Material] パラメータの右端にある [Operator Chooser] (🔿) ボタンを クリックします。ポップアップウィンドウから、brick-material に移動 して強調表示します。[Export Relative Path] オプションをオンにし、 [Accept] をクリックします。

マテリアルが割り当てられましたが、ブロックはまだレンダーされま せん。

Material Paletteに戻って brick_material を選択します。 [Surface] タブをクリックした後、[Base Color] を 0.5, 0.5, 0.5に設定し、[Use Packed Color]オプションをオンにします。ブロッ クが赤色でレンダーされるようになったはずです。

そうでない場合は、変更が適用済みであることを確認するためだけ に、Scene ビューの上部バーにある [Render] をクリックしてくださ い。これで、ポイントに割り当てられたカラーに合わせてブロックも 赤色でレンダーされるようになります。

ここまでの作業内容を保存 ([Save]) します。Render Region の右上に ある**x**ボタンをクリックして閉じます。



パート 4: ティーポットに切り替える

ここでは、ゴムのおもちゃとティーポット間の切り替えを行う switch ノードを追加します。 ノードのネットワークが異なる形状を処理しながらも望みどおりの結果を生成する様子が示されます。 これは、このネットワークのプロシージャルな「レシピ」が正常に機能していること示しています。



01 sを押して選択ツールに移動します。ネットワークビューで、 Tab キーを押して Switch という単語を入力した後、Sourcing フォルダからそれをネットワークビュー内にドラッグします。次にそれ を、rubbertoy ノードを pointsfromvolume ノードに接続しているライン の上にドラッグします。すると、ネットワーク内のその2つのノードの 間にそれが挿入されます。

このノードを使えば、異なる入力形状間の切り替えが容易になります。



Series Selectors Selectors Second Con-

The second secon

02 ツールシェルフの [Create] メニューに移動し、[Platonic] ツールをネットワークビューにドラッグします。すると、 platonic ノードがジオメトリレベルに配置されます。ツールは、その ノードタイプが現在のネットワークレベルで意味をなす限り、ネット ワークビュー内にドラッグすることができます。

platonic ノードの **[Solid Type]** を **[Utah Teapot]** に設定します。 **[Radius]** を **4** に、 **[Position Y]** を **1.9** に設定します。points from volume ノード でボリュームに合わせてポイントが更新され、ブロックの新しい構成 が得られます。



これが、プロシージャルシステムの大きなメリットの1つです。後で、 デジタルアセットと呼ばれるカスタムツール内にこのネットワークを パッケージ化します。**デジタルアセット**としてネットワークをパッケー ジ化すれば、他のユーザとネットワークを共有することが容易になる ほか、ツールの進化に伴う変更/更新が不可避のスタジオ環境に配備す る際のツールの管理も容易になります。



04 *rubbertoy* に戻すには、*switch* ノードの [Select Input] を 0 に 設定します。以上で、ブロック化の対象となる形状として、 これら 2 つの形状間の切り替えはもちろん、別の形状の追加も可能と なりました。

作業内容を保存 ([Save]) します。

パート 5: テクスチャを使ってポイントにカラーを追加する

パート3では、ブロックインスタンスのカラーに影響を与えるカラーアトリビュートをポイントに追加しました。 ここでは、単一のカラーを使用する代わりにテクスチャマップを使用して、ブロックのより興味深い外観を実現します。 その際、テクスチャをポイントカラーに変換するための特殊なノードがいくつか必要になります。



testgeometry_rubbertoyの表示フラグをオンにします。rubbertoy ノードを選択し、パラメータペインで [Add Shader] パラメー タを**オフ**にします。

ブロックのカラーは別の方法で取り戻します (具体的には、ディスク上 のテクスチャマップからカラーを取り出すことでブロックにカラーを 追加します)。



[Asset] メニューから [Edit Asset Properties] > [Rubber Toy] を選択します。Properties ウィンドウで [Extra Files] タブ をクリックし、toylowres.jpg を選択します。

[Save as File] ボタンをクリックし、それを tex フォルダ内に保存しま す。デジタルアセット内にテクスチャが格納されていたのは、テクス チャもアセットと一緒に共有できるようにするためです。ディスク上 のテクスチャを使ってブロックにカラーを追加します。



e 🗾 🔜 😑 🔍 🖻 4 248 246

ネットワークビューで Tab キーを押して Attribute... と入 カし、[Attribute VOP] ノードを選択し、ネットワーク内の pointsfromvolume ノードの横に配置します。

switch ノードの出力を attributevop ノードの最初の入力に接続します。 この新しいノードのDisplay Flag (表示フラグ) を設定します。

パラメータペインで、[Run Over]を [vertices] に設定します。

attributevop ノードを**ダブルクリック**して下のレベルに移動し、 Tab キーを使って Texture VOP を追加します。これを、 geometryvopoutput ノードの Cd 入力に接続します。 UV coordinate ノードを追加し、texture ノードの UV 入力に接続します。





* * 7

05 Texture ノードを選択します。Texture Map パラメータの 右端にある歯車 (際) アイコンをクリックし、メニューから [Promote Parameter]を選択します。すると、このノードの上のレベ ルに、このパラメータが追加されます。

mapの横に表示される小さなノブをクリックします。パラメータペイン で [Label] を Texture Map に変更します。

このチェーンの末尾に Attribute Promote を追加します。[Original Name] を [Cd] に、[Original Class] を [vertex] に設定します。[New Class] は [Point] に設定されたままにします。

OT Attribute Transfer ノードを追加します。pointsfromvolume を最初の入力に接続し、attribpromote を 2 番目の入力に接続します。[Attributes]タブで、[Points]フィールドの右側にある矢印を クリックし、[Cd] を選択します。

color ノードの後に switch ノードを追加し、この switch に *attribtransfer* ノードを接続します。このノードの名前を *texture_switch* に変更します。

[Switch]を1に設定します。copytopoints ノードの表示フラグを設定します。これで、テクスチャマップのカラーが、コピーされたポイント にトランスファーされ、したがってコピーされたブロックにもトラン スファーされるようになります。

作業内容を**保存** (**[Save]**) します。

アトリビュートトランスファー

0

あるジオメトリのアトリビュートを別のジオメトリにコピーしたい場合は、 Attribute Transfer を使用できます。このノードでは、**Distance Threshold** や その他のパラメータを使用してアトリビュートのコピーが実行されます。 ゴムのおもちゃの場合、ジオメトリ上のポイントの**Cd** アトリビュートが、 ブロックのコピーに使われるポイントクラウドへトランスファーされます。

M

パート 6: ブロック化デジタルアセットを作成する

この時点でブロック化のレシピは正常に機能しており、ノード間の相互接続も正しく行われているので、 一部のノードをラップして単一の Houdini Digital Asset [HDA] ノードを作成します。このアセットの内部に 含まれるパラメータを使ってネットワークを共有できます。これらのパラメータは、アセットが使用される たびにユニークな結果を生成できるインターフェースを作成する目的で、トップレベルへと昇格されます。

● 1 ネットワークビューで、platonic ノードをドラッグして端の ほうに移動します。ネットワーク内の他のノードをすべて選 択し、[Assets] メニューから [New Digital Asset From Selection...]を 選択します。すると、それらが単一のノードに折りたたまれます。

[Operator Name] を brickify に設定します。すると、[Operator Label] が Brickify に変更されます。[Save to Library] の右端にあるボタンをク リックします。[Location] サイドバーで \$HIP を選択した後、HDA ディ レクトリをダブルクリックします。Create New Digital Asset ダイアロ グで、[Accept] を押した後、[Accept] をもう一度クリックします。

👩 Edit Operator Type Properti

02 すると、Type Properties ウィンドウが表示されるので、 [Basic]タブの内容が表示されることを確認します。[Minimum Input]を0に設定し、入力がなくてもアセットが動作するようにしま す。許可される入力ノードの個数を定義する[Maximum Inputs]パラ メータは、1に設定されています。

これが、platonicノードに現在接続されている入力です。後でこのデジ タルアセットを使用する際には、この入力を使って別の形状を参照し ます。

[Apply]を押します。[Accept]は押さないでください。押すとウィンド ウが閉じてしまいます。

03 ネットワークビューで、新しいアセットノードの名前を brickify_asset に変更し、ダブルクリックしてその内側に入り ます。testgeometry_rubbertoyノードとSubnetworkInputノードの切り替 えを行っている switchノードをクリックします。この入力ノードは、 上のレベルからプラトン立体のティーポット形状を取得しています。

Type Properties ウィンドウで [Parameters] タブをクリックします。 Select Input パラメータの名前をクリックし、Type Properties ウィ ンドウの [Existing Parameters] リストに左マウスボタンでドラッグ します。root の上にパラメータをドロップすると、UI にそのパラメー タが追加されます。[Apply] をクリックするとパラメータが追加されま すが、Type Properties ウィンドウは閉じられません。

Inda ファイルとは

アセットを保存すると、ディスク上に.hda ファイルが作成されます。HDA は Houdini Digital Asset を表しています。アセット定義は、このファイルに格納され た後、シーンから参照されます。このファイルには、ノード、昇格されたパラメー タ、UI 要素などに関する情報が含まれています。このファイルの参照は、複数の ユーザがさまざまなショットから、共有されたエクスペリエンスのために行うこと ができます。

シーン内で参照されているアセットを管理するには、[Assets] メニューから [Asset Manager...] を選択し、[Current HIP File] を開きます。

_		1		_			
Operators	Configuration						
🖨 – Operat	or Type Libraries						
📮 - Cur							
ė -	C:/Users/rob/Do		idiniProjects/b	rickify_lesso	n/hda	/brickify	.hda
	└── Sop/brick						
🖽 – Sca							
🖬 – Interna							
- Other S	Scripted Operator						
			li -	_	_		
Filter							
Find Operat	tor Definition						
	-	_	_		_	_	_
0							

:- @

🗘 🛄 🕫 🐴 🔪 🐅 🍇 😪 🌴 🚢 Select 🔹 Visibility 🔹

Extra Files

04 ネットワークビューのパスバーの brickify_asset で右マウスボ タンをクリックし、[Parameter and Channels] > [Parameters] を選択します。すると、brickify アセットの2 つのパラメータを含むフ ローティングパラメータウィンドウが表示されます。これらは、この アセットを使用するすべてのユーザが利用できるパラメータです。他 のパラメータをいくつか追加しましょう。

brickifyノードには新しいパラメータが含まれています。その値を**0から** 1 に変更したり元に戻したりして、シーンにどのような影響が生じるか 確認してみてください。問題は、名前があまり適切ではなく、またこ の場合はスライダよりメニューのほうが格段に優れている点にありま す。そこで、Type Properties を使って UI を改善します。

05 パラメータリストで Select Input パラメータをクリックします。 左側に、その外観を改善するためのオプションがあります。 その [Name] を shape に、[Label] を Shape に変更します。

次に、[Menu] タブをクリックし、[Use Menu] をオンにします。次に、 [Menu Items] の下で、[Token] で0と入力し、[Label] で Rubber Toy と 入力した後、Enter を押します。次に、[Token] で1と入力し、[Label] で Custom Shape と入力します。[Apply] を押します。

これで、フローティングパラメータペインにメニュー付きの Shape パ ラメータが表示されるようになります。使ってみてその動作を確認し てください。

O6 color ノードと attributetransfer ノードのすぐ下にある2つ目の switch ノードを選択し、その Select Input パラメータをパラメータリストに昇格させます。その [Name] を look に、[Label] を Look に変更します。

次に、[Menu] タブをクリックし、[Use Menu] をオンにします。次に、 [Menu Items] の下で、[Token] で 0 と入力し、[Label] で Color と入力 した後、Enter を押します。次に、[Token] で 1 と入力し、[Label] で Texture Map と入力します。[Apply] を押します。

🗱 Parameter Description

🖌 Use Menu

1

Token

Use Menu Item Token as Value

Menu Items
 O Menu Script

Parameter Channels Menu Import Action Button

Normal (Menu Only, Single Sel... 🍦

Texture Map

Label

07 次に、ネットワーク内で color ノードを選択し、Color パラ メータをパラメータリストに昇格させます。すると、カラー ウィジェット、カラーホイール、および RGB フィールドが追加されます。

TypePropertiesウィンドウで[**Apply**]を押し、*brickify_asset*のパラメー タインターフェースでこのパラメータがどのように表示されるかを確 認してください。

注: [Accept] を間違って押した場合、新しいパラメータはアセットに 保存されますが、[Asset] メニューから [Edit Asset Properties...]> [brickify] を選択してウィンドウを再度開く必要があります。

)
8 次に、attributevop ノードを選択し、Texture Map パラメ-タをパラメータリストに昇格させます。

[Parameter Description] セクションの [Channels] タブをクリックし、 [Defaults] の値を Mandril.pic に変更します。これはデフォルトのテ クスチャマップであり、ロード元のパスを必要としない、信頼性の高 いデフォルトです。後で、toylowres.jpg テクスチャマップを再ロードし ます。

Type Properties ウィンドウで **[Apply]** を押すと、*brickify_asset* のパラ メータインターフェースにこのパラメータが表示され、ブロックのカ ラーが Mandril テクスチャマップから取得されるようになります。

kify brickify asset

ire Map 🛔

\$HIP/tex/toylowres.jpg

これにより、[Look] メニューで **[Color]** が選択されなかった場合は常に このパラメータが無効化されるようになります。次に、**[Texture Map]** をクリックし、**[Disable When]** フィールドに { look != 1 }と入 力します。

[Apply]を押し、[Shape]メニューを使って結果をテストします。不要 なパラメータは無効化されることがわかります。[Hide When] オプシ ョンを使って不要なパラメータを非表示にすることもできますが、こ こでは無効化が適しています。

1 brickify_asset の **Texture Map** パラメータのデフォルトは **Mandril.pic** になっていますが、これは汎用性の高いデフォ ルト値です。なぜなら、常に利用可能なテクスチャマップだからです。 おもちゃのマップに戻したい場合は、**Texture Map** の横にあるファイ ルセレクタをクリックし、先と同様に\$HIPをクリックした後、texディ レクトリの内側に入り、toylowres.jpg ファイルを選択します。

すると、このテクスチャマップがアセットで使用されるようになりま すが、デフォルトは Mandril.pic のままです。これがデフォルトでない ことは、テキストが太字になっていることからわかります。デフォル トは Mandril のままですが、このアセットインスタンスでは別のマッ プが使用されている、ということです。

ネットワークビューで、**uを押して**1つ上のレベルに戻ります。brickify_ asset ノードが選択された状態で、メインメニューから[Assets]>[Lock Asset]>[Brickify]を選択します。確認メッセージが表示された場合に は、[Save Changes]を押します。

brickify_asset ノードを**ダブルクリック**して下のレベルに降りてみると、 ネットワークがグレーアウトされ、これらのノードが保護されている ことがわかります。このアセットの操作はパラメータ経由でしか行え ません。アセットの内部に変更を加えたい場合は、アセットをアン ロックする必要があります。

12 次に、これまで行った作業が失われないように、シーンファ イルを保存([Save])します。ユーザのファイルとしては、 シーンファイルと、シーン内でアセット作成のために参照されている .hda ファイルがあります。このライブラリを使えば、このシーン内で 他のアセットインスタンスを作成したり、別のシーンにアセットを追 加したりできます。

このレッスンの次のセクションでは、ユーザのアセットをテストし、 アセットが正しく動作するか確認します。正常に動作しているアセッ トとテスト用のアセットを用意することが常に推奨されます。ユーザ の望む動作が実現できているか確認しやすくなるからです。

er 🖬 😵 💽 💽 🗢 🚭 🖉 🖉 🖓

🗱 🕅 🔍 🕕 🕐

アセットのロックとアンロック

選択されたアセットのロックやアンロックを行うには、[Assets] メニューを使用 します。アセットが**ロック**されている場合、アセットは HDA ファイルを参照して 自身の動作を決定します。アセットが**アンロック**されている場合、アクティブな 定義はシーンファイル内に存在しています。アセットをロックする際に、変更が 存在している場合、保存を確認するメッセージが表示されます。

アセットのノード上で**右マウスボタンをクリック**すると、[Allow Editing of Contents] でアセットをアンロックしたり、[Match Current Definition] でア セットをロックしたりできますが、その際には注意が必要です。保存を確認する メッセージが表示されないので、変更内容が失われる可能性があるからです。

Assets		
New Digital Asset From Selection		
Unlock Asset	►	
Lock Asset	►	🚍 Brickify
Save Asset	•	7
Edit Asset Properties	►	

0

パート 7: デジタルアセットをテストする

デジタルアセットのインスタンス化は、単一のシーンファイル内で複数回行えます。ここでは、異なるジオメトリでこのアセットを 使用し、正常に動作するかテストします。利用可能なテストバージョンを用意することは常に推奨されます。最初のアセットに施さ れた変更をすばやく検証できるからです。正常動作することが確認されたアセットは、他のシーンファイル内でも使用できます。

01 オブジェクトレベルに戻り、Scene ビューで **Tab** キーを使っ て[**Squab Test Geometry**]を選択します。**Enter**を押して原 点の位置に配置した後、ハンドルを使って端のほうに移動させ、ゴム のおもちゃとの距離を確保します。

新しいオブジェクトノードをダブルクリックしてジオメトリレベル に降ります。ノードを選択し、[**Scale**] を **3** に設定します。すると、 Squabがゴムのおもちゃより少しだけ大きくなります。

02 testgeometryノードの出力で右マウスボタンをクリックし、 brickify…と入力し始めた後、メニューからbrickifyアセット を選択します。すると、この新しいネットワーク内にアセットが配置 されます。

その**表示フラグ**を設定すると、**Mandril**テクスチャマップのカラーを持 つ別のゴム製おもちゃが表示されます。これは、これらがこのアセッ トのデフォルトだからです。

O3 brickify アセットノードで [Shape] パラメータを [Custom Shape] に設定し、イカ蟹がブロック化される様子を確認してください。新しい形状がアセット内のノードネットワークによって 処理され、ユニークな結果が生成されます。

このように、デジタルアセットは、複数のアクションを1つのノードに パッケージ化するためにパイプライン内に配置可能なツールとなりま す。このアプローチを使えば、ワークフローの作業速度が増し、より 一貫性のある結果を実現しやすくなります。

04 testgeometry_squab ノードを選択します。[Asset] メニュー から [Edit Asset Properties] > [Squab] を選択します。 Properties ウィンドウで [Extra Files] タブをクリックし、squab_ diffuse.jpg を選択します。

[Save as File] ボタンをクリックし、それを tex フォルダ内に保存しま す。デジタルアセット内にテクスチャが格納されていたのは、テクス チャもアセットと一緒に共有できるようにするためです。

次に、ディスク上のこのテクスチャを使ってブロックにカラーを追加 します (brickify ノードの **Texture Map** パラメータ経由)。

パート 8: ブロックのアニメーション

アセットに機能を追加し続けることも可能です。ここではブロックの自動構築アニメーションを作成します。 この新機能をアセットに追加するには、ネットワークにいくつかのノードを追加する必要があります。 .hda ファイルに結果が保存されると、このアセットを利用しているすべてのユーザが機能を利用できるようになります。

brickify_asset ノードを選択し、メインメニューから [Assets] > [Unlock Asset] > [Brickify] を選択します。brickify_asset ノードをダブルクリックした後、texture_switch の出力で**右マウスボタ** ンをクリックし、Group... と入力してから [Group by Range] を選択 します。ノードを配置し、その**表示フラグ**を設定した後、以下のパラ メータを設定します。

- Group Name]: hide_points
- Group Type]: [Points]
- [Method]: [Start and Length]
- [Length]: (\$F-1) *20
- [Invert Range]: オン
- [Range Filter]の下では値は次のまま: [Select]: 1、[Of]: 1

group ノードの出力で右マウスボタンをクリックし、 [Polygon] > [Blast] を選択します。このノードを配置した 後、[Group] の横にある矢印を使って hide_points グループを選択しま す。次に、[Delete Non Selected] をオンにし、グループの外側のポ イントが削除されるようにします。blast ノードの表示フラグを設定し ます。

再生ボタンを押し、フレームごとにポイントが増えていく様子を確認 してください。次に、チェーンの末尾にある material ノードの表示フ ラグを再設定し、ブロックが時間とともに増えていく様子を確認して ください。

この時点では、ブロックは地面からではなく、いずれかのサ イドからやってきています。これは、ポイントがそのポイン ト番号に基づいて表示されているからです。これを制御して望みどお りの外観が得られるようにするには、ポイントを並べ替える必要があ ります。

texture switch ノードの出力で右マウスボタンをクリックし、Sort...と 入力してから [Sort tool] を選択します。このノードを配置し、[Point Sort] を [Along Vector] に変更します。これが 0, 1, 0 に設定されてい ると、ポイントは底面から始まり、上に上がっていきます。

再生して結果を確認してください。さまざまなベクトルを試し、その 違いがアニメーションに及ぼす影響を確認してください。

ブロック化エフェクトのアニメーションを実行するかどうか を選択できるようにするには、別の switch ノードを追加し ます。ネットワークビューで Tab キーを押してSwitch... と入力し、 [Switch] を選択し、ノードを配置します。その名前を animation_switch に変更します。

texture switch ノードの出力をクリックし、それを switch ノードの入力 に接続します。blast ノードでもこれを繰り返します。これにより、元 の形状が1つ目のオプション、アニメーションエフェクトが2つ目のオ プションになります。[Select Input]を1に変更すれば、ブロックのア ニメーションが表示されますが、ここでは0のままにしておきます。

🖌 La

🖌 Rang

- 🕰

enu Import Action B build_speed

Build Sp

パスバーの brickify_asset で右マウスボタンをクリックし、 [Type Properties...] を選択します。[Parameters] タブを クリックした後、[Create Parameters by Type] タブから、Color のす ぐ後にある separator を追加します。

次に、switch ノードの Select Input を、パラメータペインから新しい セパレータのすぐ下にドラッグします。その [Name] を animate_bricks に、[Label] を Animate Bricks に設定します。次に、その [Type] を [Toggle] (オン[0]/オフ[1]の設定に限定) に変更します。

[Parameter Description] セクションの [Channels] タブをクリック し、デフォルト値を0[オフ]に設定します。

[Apply]をクリックして変更を保存します。

Type Properties の [Create Parameters] セクションから animate_bricks パラメータの下に、Integer パラメータをド ラッグします。その [Name] を build_speed に、 [Label] を Build Speed に 設定します。[Range] オプションをオンにした後、最初の値を1に、 2番目の値を 20 に設定します。1の横にある [lock] をクリックし、数 値が決して1未満にならないようにします。

[Create Parameters] セクションから separator をドラッグし、新し いアニメーションパラメータと他のパラメータを区切ります。

[Accept] を押して変更を保存し、ウィンドウを閉じます。

このパラメータはまだどの情報にも関連付けられていません が、今からこれを使って grouprange ノードの [Length] の式 を変化させます。新しい grouprange ノードを選択し、[Length] の式を (\$F-1)*ch("../build speed") に変更します。

これで、アニメーションの速度をこのパラメータで変更できるように なりました。brickify_asset ノードが選択された状態で、メインメニュー から [Assets] > [Save Asset] > [Brickify] を選択します。この方法を 使えば、Type Properties ウィンドウを再度開かなくても、.hda ファイ ルにこの式を保存することができます。このステップを忘れないよう にしてください。そうしないと、この式がアセットの定義の一部にな りません。

brickify_asset ノードが選択された状態で、メインメニューか ら [Assets] > [Lock Asset] > [Brickify] を選択します。以上 で、ブロック化エフェクトをカスタムツール内にラップすることがで きました。このカスタムツールは、さまざまなアーティストがさまざ

brickify ノードの新機能が利用可能になったイカ蟹のネットワークに移 動します。[Animate Bricks] トグルをオンにし、[Build Speed] を設 定します。なお、この形状ではブロック数が多いので、この値を 100 前後に設定しなければならない可能性があります。

再生して結果を確認してください。

0 まとめ

ン Ш ト い

× 11 6

まなショットで利用することができます。

a norma bankwan bankwan bankwan bankwan bank bank bank bank bank bank bankwan bankw

HOUDINI ENGINE 他のアプリケーションへの HDA のロード

ディスクへの Houdini Digital Asset [HDA] の保存が完了したら、そのアセットを、Houdini Engine プラグインを使って ホストアプリケーション内にロードすることができます。これらのプラグインを使えば、仲間とアセットを共有できます。 アセットは、Autodesk Maya や 3DS Max などの 3D アプリケーションに直接ロードすることもできますし、Unity や Unreal Engine などのゲームエディタにロードすることもできます。

01 これらのアプリケーションのいずれかで Houdini Engine を使用するには、まず Houdini インストーラを使ってプ ラグインをインストールする必要があります。インストールが完了 すると、プラグインが利用可能になりますが、ユーザのセッション 内で Houdini Engine を利用できるようにするには、追加の手順を 実行しなければならない場合もあります。

詳細については、次のページにアクセスしてください。

Sidefx.com/ja/products/houdini-engine/ もしくは SideFX.com/engine(日本語訳あり)

詳細については、[Engine Plug-ins] タブをクリックした後、目的の プラグインをクリックしてください。 02 プラグインのインストールが完了したら、[Houdini Engine] メニューまたは [Import] ボタン (UE4 の場合) を使ってア セットをロードできます。ロードが完了すると、ブロック化アセッ トがビューポートに表示されるとともに、アセットのパラメータが 操作用として利用可能になります。

また、[Shape] を [Custom Shape] に設定し、ホストアプリケー ション内のジオメトリにアセットを接続することもできます (その オブジェクトにブロック化が適用されます)。

また、Mayaまたは3ds Maxでアニメーションをオンにし、そのシー ケンスをタイムラインを使って再生することもできます。

Autodesk Maya

Autodesk 3ds Max

ブロックのカラーはどうなった?

さまざまなホストアプリケーション内で、ブロック化されたどの形状においてもブロックに色が付いていないことに、気づいた方がいらっしゃるかもしれません。これは、情報の処理方法が、プラグインと Houdini とで常に同じとは限らないからです。ポイントのカラー情報は依然としてアセット内に含まれていますが、ホストアプリケーションはこの情報を受け取っていません。

またアニメーションは、Maya や 3ds Max では動作しますが、Unity や Unreal Engine では無意味です。なぜなら、Houdini のアセットがゲームのランタイムエクスペリエンスの一部になることはできないからです。したがって、組み込まれている アニメーションは無視されます。ホストアプリケーションの機能に応じてアセットを調整することが重要です。特定のアプ リケーション向けのアセットを作成する方法を学びたい場合は、**PROCEDURAL GAME ASSETS FOR UE4** レッスンを確認し てください。