

Geometry Nodes in Blender 2.92の紹介-チュートリアル1/2

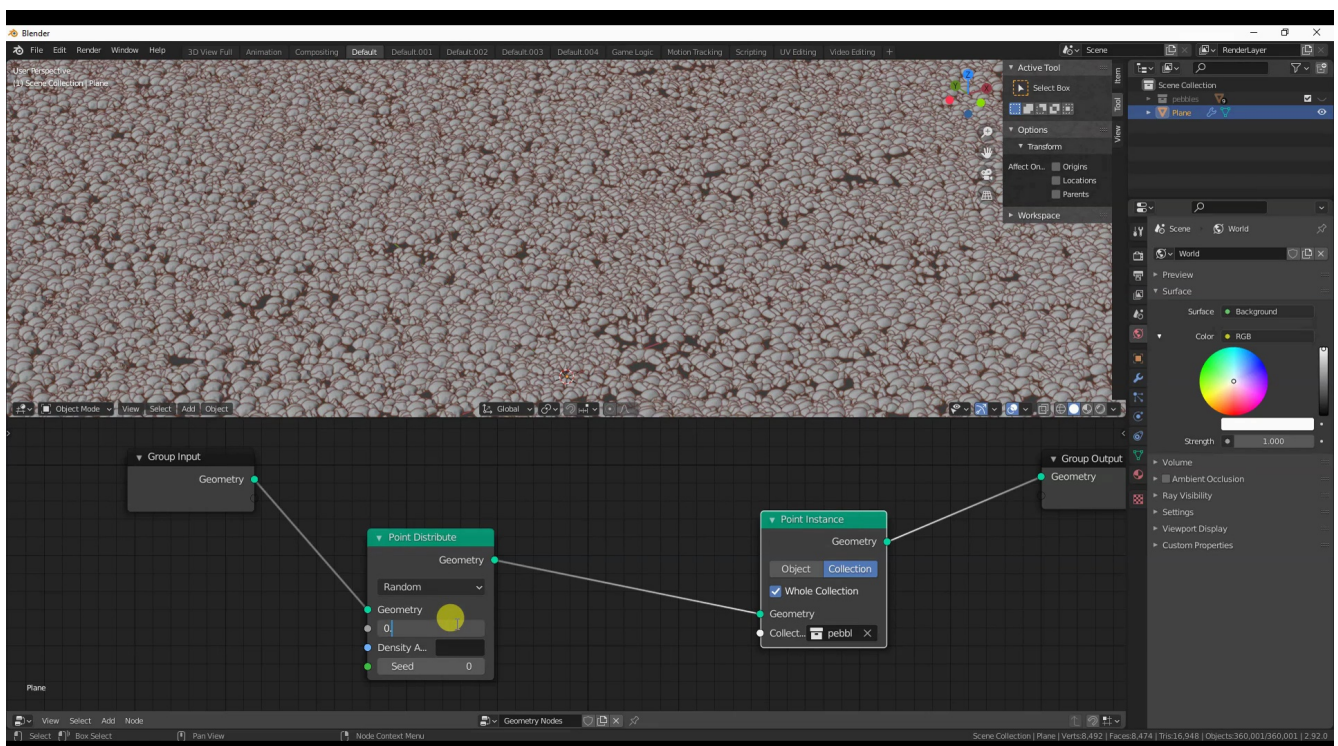


このチュートリアルで使われる3Dモデル： Office wall clock 1
(<https://sketchfab.com/3d-models/office-wall-clock-1-7fafd736d520456eac2066808b0a0006>)
(<https://sketchfab.com/3d-models/office-wall-clock-1-7fafd736d520456eac2066808b0a0006>)

こんにちは、みなさん!

これは、**Geometry Nodes in Blender** (特にソフトウェアのバージョン2.92で作られた)の2つの入門チュートリアルの最初のチュートリアルです。私は、インターフェイスとモデリング(**Object**と**Edit Mode**の両方で)に関するいくつかの情報を当然のことと考えるので、これらのチュートリアルは、すでに**Blender**の基本的な知識を持っている人のためのものです。

Geometry Nodesは、**Blender**バージョン2.92で導入されたツールで、オブジェクトをパラメトリックな方法で変換することができます、つまり、値を使用する;つまり、グループ内で複数の操作を実行し、それらを組み合わせることができるモディファイヤです。



YouTube では、**Geometry Nodes**を使用してパーティクル システムを置き換える方法を示すチュートリアルが数多くあり、表面上の岩や草の刃など、他の(または内側)にオブジェクトのインスタンスを”散乱”するため... さて、次のチュートリアルでは、私も同じ例を行いますが、主題に関するこの最初のチュートリアルでは、いくつかの基本的な概念に焦点を当てたいと思います。

Geometry Nodesはモディファイヤであり、モディファイヤは非破壊的な方法で変換を実行できるため、**Blender** 演算子と区別されることを知っています、これは、後で一部のパラメータを変更し、他のモディファイヤでも適用される変更を見つけることができることを意味します; たとえば、ビデオでは、**Array modifier** (上)で作られたコロネードと**Spin operator** (下)で作られた別のコロネードの違いを示しています: 列の高さを変更するには、**Array** の場合は、私は**Edit Mode**に行き、他のすべての要素の変化を見つけるために基本的なジオメトリを上げる必要があります; **Spin operator**を使用して、代わりに、作成されたすべての要素を削除し、**Edit Mode**で最初の要素を変更してから、**Spin**を再び適用する必要があります。

さらに、**operator**との操作が完了すると、別のツールに切り替えると、適用されるパラメータを変更できなくなりますが、**modifiers**ではこれらのパラメータは使用できなくなり、後で変更することもできます(もちろん、**modifier**を適用するまで)。

これは本当に基本的な例です、しかし、**modifiers**を使用する利点-可能な場合-特に、要素にさまざまな変更を加えるリスクがある場合、またはさらに悪いことに、連続して操作を行う必要がある場合、変更によって後続の操作をやり直す必要がある場合は特に明らかです; また、**modifiers**を使用したモデリングがパラメトリックと非破壊と呼ばれる理由も説明しています。

したがって、**Geometry Nodes**は**modifier** ですが、**Blender**の他の基本的な**modifiers**とは異なり、**nodes**を組み合わせることでカスタマイズできます。

Nodesは基本的に、2 種類の要素を視覚的に表現したものです: 情報または操作; **Blender**で材料またはポストプロダクションを作成した経験がある場合は、たとえば、入力情報 (**Node**で定義) を取得できることが分かります、(1 つ以上の中間**Node**の組み合わせを通じて) 操作を実行し、結果を出力で最終的に取得し

ます (これは**Node**で定義されます)。それでは、**Geometry Nodes**を使用して、この概念がオブジェクトの変換にどのように適用されるのを見てみましょう。

このチュートリアルの例のモデルとして、私は、壁時計を使用しています; このモデルへのリンクはビデオの説明に記載されていますが、必要はありません: あなたはチュートリアルに従って、必ずしもこのモデルでそれらを複製することなく、概念を学ぶことができます、もちろん!

Geometry Nodesを操作するための**Blender**インターフェースを設定する, 3Dビューで**"Geometry Node"**タイプエディタを開き、選択したオブジェクトの**Modifiers**タブを**"Properties"**エディタで開きます (注: ここでは、1つのモニターでビデオを録画しているので、このレイアウトを設定しています; 2つ目のモニターを使用している場合は、1つのモニターで**3D Viewport**と**Properties**エディタを使用し、もう1つのモニターで**"Geometry Nodes"**エディタを開くをお勧めします)。

分針と時針は別々のオブジェクトで、クロックに親になっており、時間針は別々のオブジェクトです; 両方のオブジェクトは、それが必要な原点を持っています (そのピボット内), したがって、各オブジェクトの**Y** (ローカル) 軸を中心に回転すると、正しい回転が行われます; とにかく、このチュートリアルでは、2つの手の一つの**Origin**を回復してみましょう (選択して、**SHIFT S**で開くことができるメニューから**Cursor to Selected**を選択すると)、シーン内に存在する3つの手が削除されます。

その後、**Cube**を追加し、もう一方の手の**Origin**を持つ; 変換はそのポイントの周りに行われるので、これは**重要**です。

元のクロックの基部にある**Cube**を親にして、時計を変換するときにも**Cube**を移動、回転、サイズ変更できるようにします。

Geometry Nodes modifierを追加してみましょう。この操作は、オブジェクトの**Modifier**メニューまたは**Geometry Nodes**エディタの**"New"**ボタンから実行できます。

デフォルトでは、“**Group Input**”と“**Group Output**”という2つの**nodes**が作成されます。入力**modifier**に渡されるオブジェクトのジオメトリを提供します(ここではポイントがすでにモデル化されており、**Origin**はこのデータから始まります: **Geometry Input**はこのデータから始まります)、出力はすべての処理の最後に返されます。この2つの**nodes**間で実行します。

nodesは、**Add**メニューからノードを選択して追加されます (ショートカット: **SHIFT A**)。このメニューでは、特に、**nodes**はさまざまなグループに分かれているため、たとえば**Mesh**グループと**Geometry**グループの**nodes**はオブジェクト全体とそのジオメトリ(**Object**と**Edit Mode**の操作)に作用し、**Point**グループのノードは生成されたパーティクルに作用します。他のノードは、オブジェクトを変換する値を提供する場合(**Inpu**グループ)、他のノードはこれらの値(**Utilities**と**Vector**グループ)を変更する役割を果たします。

キューブを分針に変換するには、**Geometry**を追加してみましょう - **Transform node, Group Input**から**Group Output**に移動するリンクの中央に配置します (**Blender**は自動的に接続を行います)。

データの流とその操作を駆動するさまざまな**nodes**の入力ポートと出力ポートの色を見てみましょう: **Geometries**は緑色で、オブジェクトのジオメトリ全体に関する情報が渡されることを示します。一方、**vector**タイプのゲート(回転軸ごとに1つずつ、数値の三つ子を含む)は紫色です。

私たちは、分針を定義するために**Translation**と**Scale**で動作します。特に、**Translation Z**を 0.049 に設定し、**Scale XYZ**をそれぞれ 0.001、0.001、0.050 にします。

私はすぐに1つのことを指摘したい:ローカル**Y**軸の周りの回転は、分を定義するために、ここで移動し、サイズ変更されたジオメトリに関して行われる操作のように、この**node**で実行すべきではありません。

立方体の**Origin**を中心に回転するには、新しい**Geometry**を追加する必要があります-作成したノードと**Group Output**の間の変換ノード: そこで**Y**軸を中心に回転することができます。

もう一つ注意すべき点は、これらの変換が行われることです, いわば, "モディファイヤーの内側": **Item**の検査- 3D ビューのオブジェクトの**Transform**タブ, 実際には、**Location**、**Rotation**、**Scale**は元のオブジェクトには適用されなかったことに注意してください。さらに、3D ビューで**Edit Mode**に切り替えると、実際にはキューブがまだ存在し、元の特性を持っていることに気付きます。

この動作, **Subdivision Surface**などで見つかったものとすべての点で類似しています, **Geometry Nodes**が実際には**modifier**であることを理解させる(実際には、**Modifiers**タブで適用できます, したがって、変換を決定的にする... しかし、今、それを適用しないでください)。

Geometry Nodeエディタに戻って、最初の**Transform node**を選択し、複製します(**SHIFT D**を使用して), 次に、**Group Input**の**Geometry**出力をこの新しい**node**の**Geometry input**にドラッグし、この**node**の出力を**Group Output**のジオメトリ入力に渡します。

node内では、**Scale Z**を 0.025 に、**Translation Z**を 0.025 に変更し、時間針を取得します。

ご覧のとおり, オブジェクトの初期ジオメトリを取得することが可能です(キューブ この場合は) **Group Input**から複数のオペレータに渡す... しかし、同じ**Object**内の両方の修正されたジオメトリを取得する方法は?

それは簡単です: **Geometry**を使用する - **Join Geometry node**, これにより、複数のジオメトリを1つにマージし、その結果を他の**nodes**に渡すことができます; したがって、**Edit Mode**の**Join operator**は何を行うかですが、ここでは**modifier**内の非破壊的な方法で使っています!

分針の回**Rotation Y**値を 0 にリセットし、このノードを複製して、一瞬前に作成した時間針のワークフローにコピーを配置します: このようにして、時間の両方に対して、分の回転角度を個別に設定できます。

そこで、**Geometry Nodes**を使用して小さなインタフェースを作成し、**Modifier**タブで時間と分をすばやく設定し、手を自動的に動かす方法を見てみましょう。

実行したいのは、時間に基づいて**Rotation Y**を設定することです (分の場合は 0 ~ 60 の数値、時間は 0 ~ 12 の数値です)。分の**Rotation**から始めましょう。

Rotationには紫色のエントリーポートがあることに注意してください、ベクトルなので; 入力チャンネルを分離するため、ベクトル - **Combine XYZ node**を結合し、**Transform**の**Rotation**入力に追加された**node**の**Vector**出力を接続します、次に、**Combine XYZ**の**Y**入力をグループ入力の新しい入力に接続します (**Combine XYZ**から**Geometry**の下**Group Input**の半可視入力ポートにリンクをドラッグします)。

Geometry Nodes modifierタブで、, “**Value**”入力パラメータがデフォルト値と共に表示されます; このパラメータの名前を "**minutes**" に変更するには、"**Geometry Nodes**"エディタの**Sidebar**を開きます(ショートカット: **N**, マウスカーソルがそのエディタにある間), その後、名前を "**Value**" から "**Minutes**"に変更します。

また、**Min**値 (ユーザーが入力できる最小値) の値**0.0**と**Max**の 60.0 も設定します。

しかし、0 ~ 60 の値を入力すると、分針が適切に回転しないことに気付きます。これは、角度**Y**の値がラジアンで表現され、特に丸角が 6.28 (つまり 2 倍の π) の値であるためです。

このマッピングを実行するには(つまり:範囲 0 ~ 60 の値を 0 ~ 6.28 の範囲に取り込む), 我々は、**Utilities**を追加します - **Group Input**の**"Minutes"**出力と**Combine XYZ**の**Y**入力のための**Map Range node**, 次に、次の方法で**node**の値(かなり直感的な意味を持つ)を設定します:

- From Min = 0;
- From Max = 60;
- To Min = 0;
- To Max = 6.28.

今, **Modifier**タブでは、時計に表示される分の値を簡単に入力でき、分針は一貫して回転します: **Y** 軸を中心にオブジェクトを回転させるよりも便利です!

時計に関しては, **nodes**のパターンは実質的に同一である, **Map Range**ノードのマッピング値を除く; だから、このように進みましょう:

- 我々は選択し、エディタの空のスペースにコピーを配置し、分の**Map Range**と**Combine XYZ**を選択して複製します。
- 新しい**Combine XYZ**の**Vector**を時計の**Rotation**入力に接続します;
- 新しい**Map Range**の**Value**入力を新しい**Group Input**出力に接続します(特に**Hours**に名前を変更できます);
- 60 ~ 12 の**Hours**の**Map Range node**の**"From Max"**値を変更してみましょう.

注意: **Group Input**を選択し、**Minutes**と**Hours**の入力ポートの**Max**値がそれぞれ60と12に正しく設定されていることを確認しましょう!

この時点で、**Blender**インターフェイスで手の時間と分を設定し、必要な時間を指定できます. 私たちは終わったのですか? まあ、いいえ: 実際の時計では、分針が前進するにつれて、時計も(1時間から次の時間に)移動しますが、これはこのモデルでは起こりません(例えば**"2:50"**または**"4:55"**を設定してみてください: 時計針は2または4で停止し、3または5に近いはずです)。

必要なのは、小さな増分、特に30度(分針が60に近い場合は12分の1回)でなければならない時計の回転の**"offset"**です。

したがって、まず、**Minutes**入力値を取り、**Utilities - Math node**で行うことができる2で**Divide**、モードを分割して"**Minutes**"を最初の値に、2番目の入力値に設定する必要があります。

得られた結果は、時計に供給される回転オフセットです、しかし、この値は度で表されますが、ラジアンで表します... まあ、我々は簡単に別の**Utilities**を紹介してそれを変換することができます - **Math node**, 今度は"**To Radians**"!

この値を時計のマップ範囲から来る値に加算するには(すでにラジアンに入っている), **Utilities**を追加しましょう - **Math node**, 今回は**Add**モードで, **Map Range**と"**To Radians**",の出力を追加する, 次に、操作の結果を**Combine XYZ**に渡しますそして、最後に、時計の**Y**回転。

理論的には、この時点で、**Modifier**タブのインターフェースを使用していくつかのテストを行うことでわかるように、すべてが機能します; さらに、クロックを変換すると、変更は手にも一貫して適用されます... またはより良い: これらは**Cube**オブジェクトに適用されます!

実際には、2つの手は同じ元の**Cube**のジオメトリであり、まだそこにあり、クロックに関連付けられています(子として): すべての変更は**Geometry Nodes node**で行われ、**modifier**を適用するまで後で変更できます。

ただし、このチュートリアルを閉じる前に、**modifier**タブの"**Hours**"と"**Minutes**"フィールドは、10進数値も受け入れ、計算で考慮されるため、視覚的な結果は正しくない可能性があります (0 を分で 9.30 時間だけ置く例を試してみてください)。

この問題を解決するためのエレガントだが機能的な方法は、**Utilities**を挿入することです - **RoundingのMath nodes - Floor**モード(つまり、最も近い下限の整数に丸める), **Group Input**の"**Minutes**"および"**Hours**"ポートの直後。これは、モディファイア インタフェースの小数点を隠すのではなく、少なくとも計算では無視します。

ここでは、今、私たちは本当に私たちの時計のインターフェイスで完了しています... また、**Blender 2.92**の**Geometry Nodes**に関するこの最初のチュートリアルでも! 次の例 (最後に、このミニシリーズ) では、サーフェスにオブジェクトを散乱させる例を見ていきます。

じゃあまたね!