

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



このチュートリアルで使用する3Dモデル：

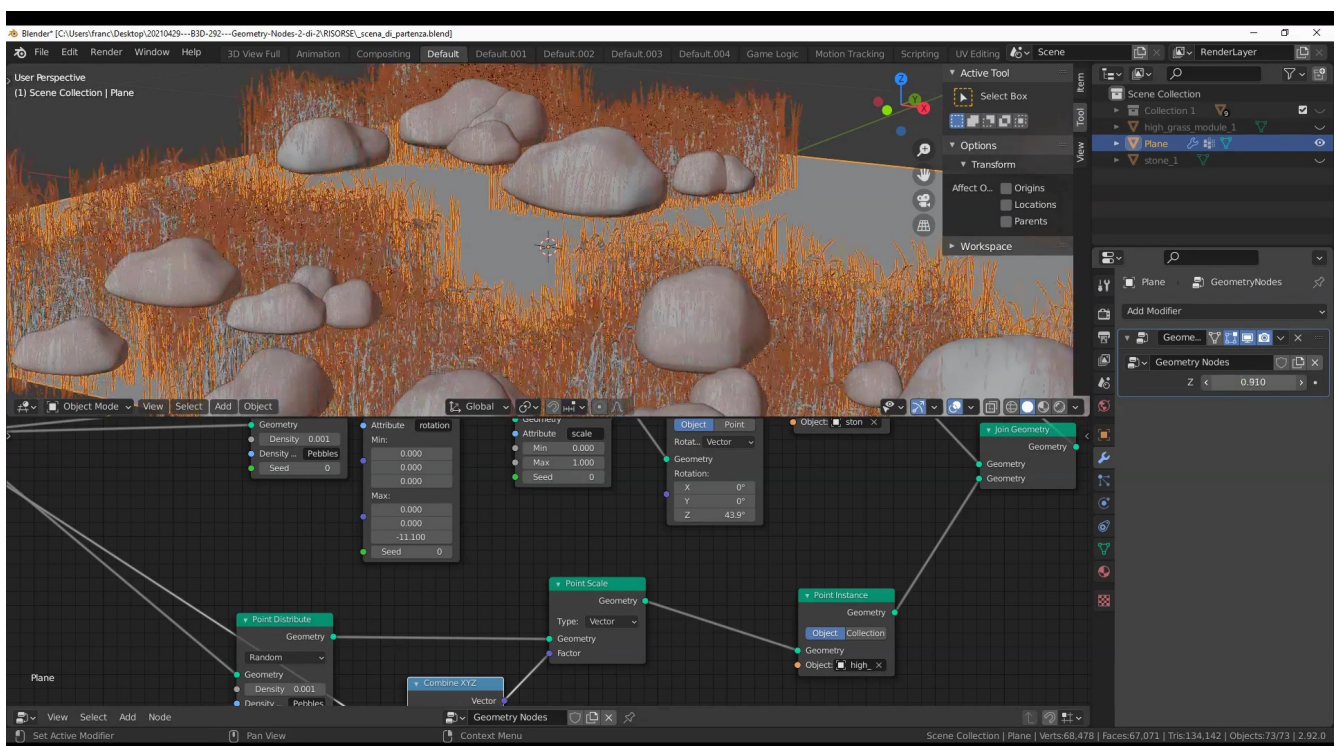
- Pebbles and stones set (<https://sketchfab.com/3d-models/pebbles-and-stones-set-8356674252f64a15ae329e1ecc78beeb>)
- High grass module (<https://sketchfab.com/3d-models/high-grass-module-e218da5203924b5d9ebd145ab0b11b2d>)

こんにちは、みなさん！

これは、**Geometry Nodes modifier in Blender 2.92**に関する 2 つの入門ビデオ チュートリアルの 2 番目です。

前のチュートリアルでは、**Geometry** グループの**nodes**を調べました(ieモディファイヤ内のジオメトリを修正およびマージするには、**Transform**と**Join**を行います); モディファイヤ タブでパラメータを指定するために、**Group Input**用の入力ポートを作成する機能, およびいくつかの**Vector**および**Utilities nodes**の使用(Math, 特に) 数学的演算を実行し、情報を結合します。

このビデオでは、**Geometry Nodes**を”散乱”のパーティクルシステムとして使用するため、別のジオメトリノードの使用、つまり他のオブジェクトの表面上のオブジェクトのインスタンスの分布が表示されます。ただし、このミニシリーズの最初のビデオチュートリアルで説明されているすべてのトピックは、ここで役に立つでしょう。



**Geometry Nodes** モディファイヤを使用すると、それを使用するオブジェクトのジオメトリをポイントジェネレータに変換できます, 表面に分布します(すなわち: その顔に).

これらのポイントは、他のオブジェクトのインスタンスを配置するために使用できます; さらに、**Geometry Nodes**を使用すると、これらの各ポイントに変換(位置、回転、サイズ変更)を割り当てて、ランダム性を導入することもできます。

これらの操作をリストアップするに当たって、私は特に3つのことを言いました:

1. ジオメトリをサーフェス上の一連の点に変換します;
2. これらのポイント内のオブジェクトのインスタンス化;
3. 生成されたポイントの特性(すなわち属性)を変更し、作成されたインスタンスに影響を与えます。

これらの3つの操作は、ジオメトリノードの数に変換され、私たちはすぐに見ます。

画面に表示されるシーンには、非常に単純なデフォルトの**Plane**があります(したがって、わずか4つの頂点と1つの面によって作られています); さらに、**APPEND**を使用して、私はいくつかの小石(the “Pebbles”)、石(“Stone 1”)と草の刃を含むオブジェクトの3Dモデルをインポートしました。

便宜上、小石を**Collection**にグループ化し、これらのアクセサリ要素をすべて隠し、**Outliner**エディタでオフにしています。

これらのアセットへのリンクは、ビデオの説明に記載されています。しかし、前のビデオの時計の場合のように、ビデオに従って、学ぶためにこれらのモデルを取得する必要はありません!

**Plane**を選択し、オブジェクトに**Geometry Nodes modifier**を追加します。

最初の操作を実行するには、平面をサーフェス上の一連の分散ポイントに変換するには、グループの中央に**Point DISTRIBUTE node**を追加します。

画面上では、**Plane**が消え、一連の点に置き換えられたことがわかります... まあ、実際には、最初の情報が失われていない、確かにそれはまだそこにあります: **Geometry Nodes**のモデリングの非破壊的な方法では、**Geometry Join node**を**Point Distribute**と**Group Output**の間に追加し、2つの**Join**入力のいずれか、**Geometry Input** (元の**Plane**)から到着するジオメトリを接続することができます。

このようにして、元の**Plane** (たとえば、地形の外観を与えるために修正やテキスト化が可能)と、その表面に分布するポイントの両方があります。

**Point Distribute node**には、**Density**が際立っているパラメータがいくつか用意されており、サーフェス上に分布するポイントの量を直感的に増減できます。

前のビデオ チュートリアルでは、モディファイヤ インターフェイスの値を変更できるように、**node**情報を**Group Input**に接続する方法を説明しました。**Point Distribute Density** パラメータに対してこれを行うのも意味があります。

次に、上記の 3 つの操作の 2 番目の操作に進めます、別のオブジェクトをインスタンス化する(またはオブジェクトのコレクション)、**Point Distribute**によって作成された各ポイントにコピーを配置する: 次に、**Point Distribute**と**Join**の間に**Point INSTANCE node**を挿入します。

操作の意味は明確です: **Distribute**はポイントを生じ、**Instance**はこれらのポイントから情報を取得して別のオブジェクトのコピーを作成します... はい、しかし、どのオブジェクト?

**Point Instance node**には、単一のオブジェクトまたはオブジェクトの**Collection**のどちらをインスタンス化するかを定義できるセレクトがあります; 選択されたモード(**Object**, 私の場合は), **node**の内側にある空のボックスをクリックして、私たちに興味のあるオブジェクトを選択することができます(**"Stone 1"**, 私の場合は).

重要な注意: 正しい結果を得る、インスタンス化するオブジェクトの**Rotation**と**Scale**変換を適用していることを確認します;これを行うには、選択して**CTRL A**キーを押し、画面に表示される**Apply**メニューから**"Rotation and Scale"**を選択します。

次に、必要に応じて**Point Distribute**の**Density**パラメータを調整し、適切な結果が得られるまで調整します。

第3の基本的な操作は、さまざまな点のパラメータを修正することです(そして 生成されたインスタンス); 実際には、さまざまなタイプの操作を実行することが可能です、そのため、単一の**node**ではなく、**Point**と**Attribute**グループのノードについて話しています。

どちらの場合も、変換ノードは**Point Distribute**と**Point Instance**の間に挿入されます、作成されたポイントを変更する必要があるため、それらのポイントからインスタンスを生成する前に行います。

たとえば、すべてのポイントに適用する回転から始めましょう: **Point Distribute**と**Point Instance**の間に**Point Rotate node**を追加し、パーティクルの回転角度を変更します。

結果は、この**node**なしで得られたものよりもすでに優れていますが、回転にランダム性を導入する方法と、なぜ、これらのインスタンスのサイズにも?

**Attribute Randomize node**に助けられます(もちろん**"Random"**,から), **Point Rotate**と**Point Instance**の間でカスケードできます(実際には、この処理フローは、さまざまな**nodes**間のリンクされた**Geometry**ポートを観察することによって確認できるため、常にジオメトリ上で行われます)。

**Attribute Randomize**は、ポイントの属性にランダム性を導入できる**node**です, **node**の**"Attribute"**フィールドに正確に指定されます。

私たちが持っている粒子の基本的な属性の中で、言うまでもなく、**"rotation"**と**"scale"**、それでは、まずこのフィールドに**"rotation"**を書き込み、**Max**値を変更しましょう...

ノードは実際にはパーティクルの方向にランダム性を導入しますが、これらはすべての軸を中心に回転しますが、これは正確には興味深い結果ではありません: **Point Rotate**で指定されたベース方向を維持し、垂直軸の周りでのみインスタンスをランダムに回転させたいと考えています。

この問題を解決するには、操作値のタイプを**Float** (3つのXYZ軸で一様に作用するコンマを持つ数値) から**Vector**に変更する必要があります, ここで、変更する軸の**MAX**を除くすべての値に**0**を設定できます...

しかし、この操作でさえ、問題をもたらします: 私たちは**Point Rotate**によって与えられた回転を失っています!

これは、ある意味で優先順位の問題です: **Point Distribute**と**Point Instance**の間では、実際には、まず**Attribute Randomize**してから、**Point Rotate**を配置する必要があります。

**Attribute Randomize**を複製し、元の**Point Rotate**との間に追加して、**Scale**パラメータを変更します(回転と同じように、完全に小文字で書かれると、**Blender**はそれを認識しません。).

ただし、この場合、すべてのパラメータ最小と最大XYZを修正して、興味深い結果を得る必要があります (また、最小値に**0**を設定しない場合は、オブジェクトは1つ以上の次元を失います)。

**Scale** 属性に関しては、最小スケール係数に1つだけ、最大値に1つのパラメータのみを使用するように、**Float**タイプに戻る方が便利な場合があります...選択はあなたです。

これら2つの属性ノードだけで、しかし、1つの幾何学で構成されたオブジェクトからある程度の多様性を得ています: **Stone 1**.

**Point Instance**でインスタンス化するオブジェクトの型を**Object**から**Collection**に変更し、小石のコレクションを指定すると、物事ははるかに興味深いものになります!

ただし、**Object**を**Point Instance**の**Stone 1**で再設定して、**Geometry Nodes**で利用できる別の機能を確認してみましょう。

古典的なパーティクル システムでは、ジオメトリのどの部分にパーティクルを作成し、どの部分にパーティクルを作成するかを定義することもできます; それは、特に、**Weight Painting**で、表面の様々な部分に(0から1まで)密度値を定義することによって行われます... まあ、私たちもここでそれを行うことができます!

ただし、先に進む前に、重み付けペイント(**Geometry Node**の外側、元のジオメトリで操作)が頂点に作用するため、**Plane**の 4 つの基本的な頂点では不十分であるため、**Plane**のサブディビジョンの数を増やす必要があります。

3D ビューでは、**Edit Mode**に入り、サーフェス全体をかなりの回数だけ細分化します。それでは**Weight Paint mode**に切り替えましょう。

デフォルトでは、サーフェスは青で表示されます, ie “重量 0”; **Point Distribute node**は、ウェイト 1 のエリアにポイントを作成します, インスタンスを表示する領域にペイントする必要があります。

注意: 特定の場所を除くすべての場所にインスタンスを適用する場合は、逆に進む必要があります; この場合、特に、**Point Distribute**から除外する領域でブラシを使用します(**Blender**は、デフォルト名として**“Group”**を持つ**Vertex Group**を自動的に作成しますが、**"Pebbles"**のようなより重要な名前に変更することもできます)。

操作の終わりに、3Dウィンドウで**Weights**メニューを開き、**“Invert”**をクリックして、ウェイトを反転します: 現在、ペイントした領域のウェイトは 0 ですが、残りのジオメトリではウェイト 1 になります。

明らかに、**Weight Paint mode**の**Tool**タブの**Weight**パラメータの値を変更し、サーフェスをブラッシングすることで、後でウェイトを変更することを妨げるものは何もありません。0 重み付き領域にインスタンスを挿入しないことを覚えておく必要があります。

**Geometry Nodes**エディタで、**Point Distribute node**の**Density**フィールドに**Vertex Group**の名前を入力します(**"Pebbles"**、私たちの場合。大文字と小文字に注意を払う): 元のサーフェス上のポイントの分布は、私たちが設定した**Vertex Weights**に比例します。

このビデオチュートリアルを終える前に、**Blender 2.92**の**Geometry Nodes**ミニシリーズの概要を説明する前に、オブジェクトのインスタンス化のさらなる例を見てみましょう....しかし、もう一つの変更で。

定義したスキームの中で、新しい**Point Density node**を挿入し、**Group Input**の**Geometry**ポートに接続する、次に、新しい(空の) **Point Instance**に接続し、シーンで使用可能な 2 つの**Point Instance**を新しい**Join Geometry node**に接続します, このノードは、シーン内で既に使用可能な**Join node**にリンクされま

す。実際には、結果のジオメトリは、2つの**Instance nodes**によって到着した1つに結合された元のジオメトリによって与えられます。

新しい**Distribute node**では、同じ**Density**属性“**Stone 1**”を設定しますが、**Point Instance**では、インスタンス化するオブジェクトとして**High Grass**オブジェクトを設定します。

必要に応じて、**Point Rotate node**を挿入して、草の刃を正しく向けることができます。

また、**Point Scale node** (**Point Rotate**と**Point Instance**の間)を挿入し、特に**Vector**モードを設定して、ベクトルで草のサイズのスケール係数を指定できるようにします。

**Combine XYZ node**を挿入し、その**Vector**出力を**Point Scale**の**Factor** (ベクトル)入力に接続してみましょう;草の刃が消え、**Combine XYZ**パラメータの既定値は0.0なので、XYを1.0に戻します (場合によってはこれらの寸法に対して適切な値を返します)。

次に、**Combine XYZ**の**Z**入力を新しい**Geometry Input**に接続します;あなたが推測したように、意図は、モディファイアインターフェイスで草の刃の高さを調整することです... しかし、なぜ自分自身を定数値に制限するか、手動でこの値を変更する必要があるのでしょうか？

**Blender**ウィンドウに**Timeline** エディタを追加し、アニメーションのフレーム範囲を設定します。例: 240フレーム。

フレーム1に行きましょう, **Z**パラメータに0.0を設定します (クロックハンドの名前を使用して前のエピソードに示されているように、その名前を変更できます); このパラメータの値を右クリックし、アニメーションキーフレームを記録するには“**Insert Keyframe**”を選択します。

アニメーションのフレーム240に移動し、**Z**の値を必要に応じて変更し、アニメーションの最後のフレームにこのパラメータのキーフレームを挿入してみましょう。

**Timeline**で**Play**を押して、アニメーションの結果を見るだけです。

まあ、それはあまりにも、このチュートリアルのためのすべてです！

また近いうちにお会いしましょう！