

Einführung in Geometry Nodes in Blender

2.92 - Tutorial 2 von 2



Das in diesem Tutorial verwendete 3D-Modell:

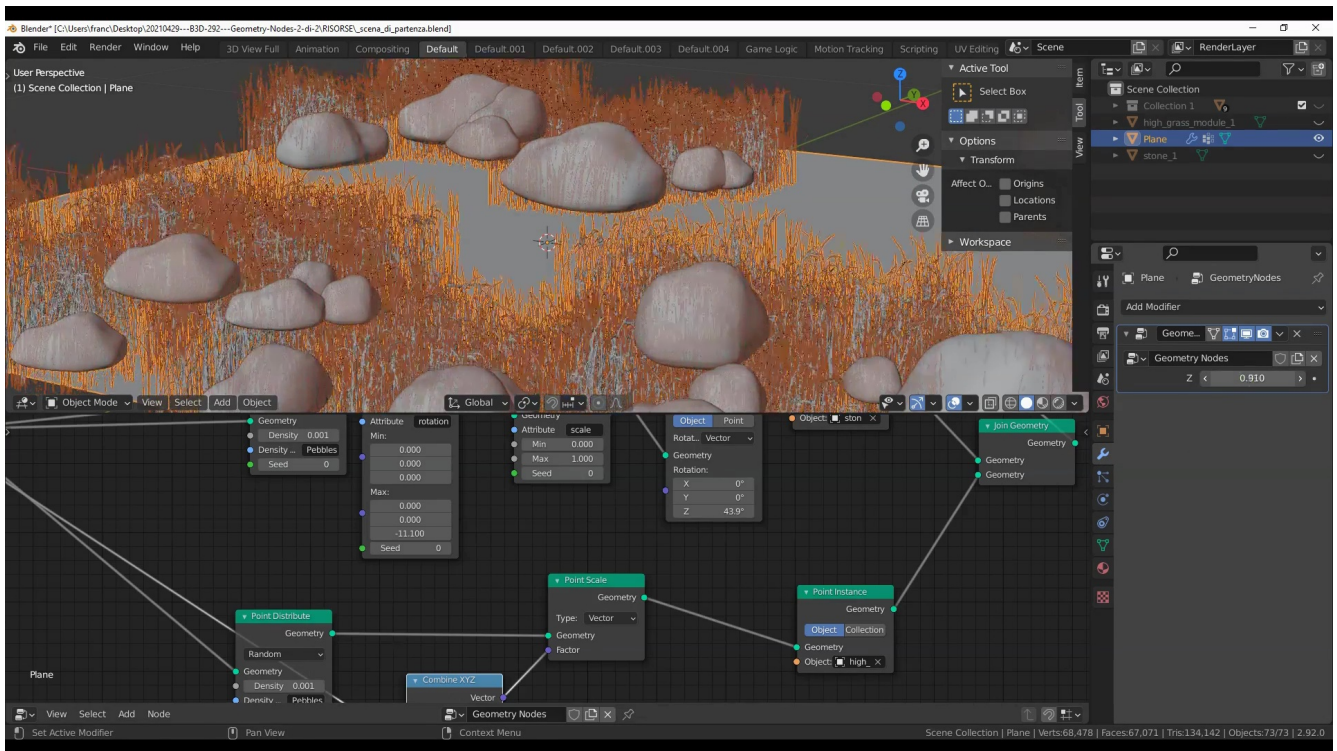
- Pebbles and stones set (<https://sketchfab.com/3d-models/pebbles-and-stones-set-8356674252f64a15ae329e1ecc78beeb>)
- High grass module (<https://sketchfab.com/3d-models/high-grass-module-e218da5203924b5d9ebd145ab0b11b2d>)

Hallo zusammen!

Dies ist das zweite von zwei einführenden Video-Tutorials über den **Geometry Nodes Modifier** in **Blender 2.92**.

Im vorigen Tutorial haben wir die **Nodes** der Gruppe **Geometry** untersucht (d.h. **Transform** und **Join**, um die Geometrien innerhalb des Modifikators zu modifizieren und zu verschmelzen). Die Fähigkeit, Eingabeports für **Group Input** zu erstellen, um Parameter im Modifikator-Tab zu spezifizieren und die Verwendung einiger **Vector-** und **Utilities Nodes** (insbesondere **Math**), um mathematische Operationen durchzuführen und Informationen zu kombinieren.

In diesem Video werden wir eine andere Verwendung von **Geometry Nodes** sehen, da wir sie als Partikelsysteme für die "Streuung", d.h. die Verteilung von Instanzen von Objekten auf der Oberfläche anderer Objekte, verwenden werden. Jedoch werden alle Themen, die im ersten Videotutorial dieser Miniserie behandelt wurden, hier nützlich sein.



Der Modifikator **Geometry Nodes** ermöglicht es uns, die Geometrie des Objekts, das ihn verwendet, in einen Generator von Punkten umzuwandeln, die auf seiner Oberfläche (d.h. auf seinen Flächen) verteilt werden.

Diese Punkte können verwendet werden, um Instanzen anderer Objekte in ihrer Korrespondenz anzuordnen; außerdem erlaubt der **Geometry Nodes** Modifikator, jedem dieser Punkte Transformationen (Position, Rotation und Größenänderung) zuzuweisen, um Zufälligkeit einzuführen.

Bei der Auflistung dieser Operationen habe ich vor allem drei Dinge erwähnt:

1. Umwandlung der Geometrie in eine Gruppe von Punkten, die auf der Oberfläche VERTEILT werden;
2. INSTANZIEREN der Objekte an diesen Punkten;
3. die Eigenschaften (d.h. die ATTRIBUTES) der erzeugten Punkte zu ändern und damit die erzeugten Instanzen zu beeinflussen.

Diese drei Operationen werden in viele **Geometry Nodes** übersetzt, die wir sofort sehen.

In der auf dem Bildschirm sichtbaren Szene gibt es eine sehr einfache Standard-**Plane** (bestehend aus nur 4 Eckpunkten und einer Fläche). Außerdem habe ich mit **APPEND** die 3D-Modelle einiger Kieselsteine (die

"Pebbles"), eines Steins ("**Stone 1**") und eines Objekts mit Grashalmen importiert.

Der Einfachheit halber fasse ich die Kieselsteine in einer **Collection** zusammen und blende alle Zubehör-Elemente aus, indem ich sie in einem **Outliner**-Editor ausschalte.

Sie finden den Link zu diesen Assets in der Beschreibung des Videos; wie im Fall der Uhr im vorherigen Video ist es jedoch NICHT notwendig, diese Modelle zu verwenden, um dem Video zu folgen und daraus zu lernen!

Ich wähle die **Plane** aus und füge dem Objekt einen **Geometry Nodes Modifier** hinzu.

Um die erste Operation auszuführen, d.h. die Ebene in eine Reihe von **DISTRIBUTE**-Punkten auf ihrer Oberfläche umzuwandeln, füge ich einen **Point DISTRIBUTE Node** in der Mitte der Gruppe hinzu.

Auf dem Bildschirm können wir sehen, dass die **Plane** verschwunden ist und durch einen Satz von Punkten ersetzt wurde ... nun, in Wirklichkeit ist die Ausgangsinformation nicht verloren gegangen, sie ist immer noch da: Die nicht-destruktive Art der Modellierung von **Geometry Nodes** erlaubt es uns, einen **Geometry Join Node** zwischen dem **Point Distribute** und dem **Group Output** hinzuzufügen und die Geometrie, die vom **Geometry Input** kommt (d.h. die ursprüngliche **Plane**), mit einem der beiden **Join**-Inputs zu verbinden.

Auf diese Weise haben wir sowohl die ursprüngliche **Plane** (die wir modifizieren und texturieren können, um ihr z. B. das Aussehen eines Geländes zu geben) als auch die auf ihrer Oberfläche verteilten Punkte.

Der **Point Distribute Node** bietet einige Parameter, unter denen die **Density** hervorsteht, mit der wir intuitiv die Anzahl der auf der Oberfläche zu verteilenden Punkte erhöhen oder verringern können.

Im vorangegangenen Videotutorial haben wir gesehen, wie man die Informationen eines **Node** mit der **Group Input** verbindet, um die Werte in der Modifikator-Schnittstelle zu ändern. Es kann sinnvoll sein, dies auch für den Parameter **Point Distribute Density** zu tun.

Wir gehen dann zur zweiten der drei oben aufgeführten Operationen über, nämlich zum INSTANZIEREN eines anderen Objekts (oder einer **Collection** von Objekten), indem wir eine Kopie für jedem durch **Point Distribute** erzeugten Punkt platzieren: Wir fügen dann einen **Point INSTANCE Node** zwischen **Point Distribute** und **Join** ein.

Der Sinn der Operation ist klar: **Distribute** erzeugt die Punkte und **Instance** nimmt die Informationen von diesen Punkten, um Kopien eines anderen Objekts zu erstellen ... ja, aber welches Objekt?

Im **Point Instance Node** gibt es einen Selektor, mit dem wir festlegen können, ob wir ein einzelnes Objekt oder eine **Collection** von Objekten instanziierten wollen. Den gewählten Modus (**Object**, in meinem Fall) können wir durch Anklicken des leeren Kästchens innerhalb des **Node** festlegen und das Objekt auswählen, das uns interessiert ("**Stone 1**", in meinem Fall).

WICHTIGER HINWEIS: Um ein korrektes Ergebnis zu erzielen, müssen Sie sicherstellen, dass Sie die **Rotation**- und **Scale**-Transformationen der Objekte, die Sie instanziierten möchten, angewendet haben. Dies können Sie tun, indem Sie sie auswählen und **CTRL A** drücken und dann "**Rotation and Scale**" aus dem Menü Anwenden wählen, das auf dem Bildschirm erscheint.

Anschließend passen wir den Parameter "**Density**" von "**Point Distribute**" nach Belieben an, bis wir ein Ergebnis erzielen, das uns gefällt.

Die dritte grundlegende Operation besteht darin, die Parameter der verschiedenen Punkte (und damit der erzeugten Instanzen) zu ändern. Es ist sogar möglich, verschiedene Arten von Operationen durchzuführen, so dass wir

nicht von einem einzigen **Node** sprechen, sondern von den Knoten der Gruppen **Point** und **Attribute**.

In beiden Fällen werden die Transformations-Knoten zwischen **Point Distribute** und **Point Instance** eingefügt, da sie die erstellten POINTS verändern müssen, BEVOR sie die Instanzen aus diesen Punkten erzeugen.

Beginnen wir zum Beispiel mit der Rotation, die auf alle Punkte angewendet werden soll: Wir fügen dann einen **Point Rotate Node** zwischen **Point Distribute** und **Point Instance** ein und ändern die Rotationswinkel der Partikel.

Das Ergebnis ist bereits besser als das ohne diesen **Node**, aber warum kann man nur die Rotation und nicht auch die Größe dieser Instanzen zufällig gestalten?

Hier hilft uns der **Attribute Randomize Node** (natürlich von "**Random**"), den wir zwischen **Point Rotate** und **Point Instance** kaskadieren können (tatsächlich findet dieser Verarbeitungsfluss immer auf den Geometrien statt, wie man anhand der verknüpften **Geometry**-Ports zwischen den verschiedenen **Nodes** überprüfen kann).

Attribute Randomize ist ein **Node**, der es uns ermöglicht, Zufälligkeit in ein ATTRIBUTE der Punkte einzuführen, das im Feld "**Attribute**" des **Node** genau angegeben wird.

Zu den grundlegenden Attributen eines Teilchens gehören natürlich "**Rotation**" und "**Scale**", also schreiben wir zunächst "**Rotation**" in dieses Feld und ändern den **Max**-Wert ...

Der Knoten lässt die Ausrichtung der Partikel durch Zufall entstehen, aber diese drehen sich um alle ihre Achsen, was nicht gerade ein interessantes Ergebnis ist: Wir möchten die durch **Point Rotate** gegebene Grundausrichtung beibehalten und die Instanzen nur um die vertikale Achse zufällig drehen.

Um dieses Problem zu lösen, müssen wir den Typ des Operationswerts von **Float** (ein numerischer Wert mit Komma, der gleichmäßig auf die drei **XYZ**-Achsen wirkt) zu **Vector** ändern, wo wir 0 für alle Werte außer dem **MAX** der Achse, die wir ändern wollen, eingeben können ...

Aber auch diese Operation bringt ein Problem mit sich: wir haben die Drehung verloren, die durch **Point Rotate** gegeben ist!

Es handelt sich in gewissem Sinne um ein Prioritätsproblem: Zwischen **Point Distribute** und **Point Instance** müssen wir nämlich zuerst **Attribute Randomize** einsetzen und dann den **Point Rotate**.

Duplizieren Sie nun das **Attribute Randomize** und fügen Sie es zwischen dem Original und **Point Rotate** ein, um auch den Parameter **Scale** zu ändern (der, genau wie Rotation, komplett in Kleinbuchstaben geschrieben werden muss, da **Blender** ihn sonst nicht erkennt).

In diesem Fall müssen wir jedoch alle Parameter min und max **XYZ** ändern, um interessante Ergebnisse zu erhalten (und vielleicht vermeiden, 0 für die Mindestwerte zu setzen, da die Objekte sonst eine oder mehrere Dimensionen verlieren).

Was das Attribut **Scale** betrifft, ist es vielleicht bequemer, zum Typ **Float** zurückzukehren, so dass Sie nur einen Parameter für den minimalen Skalierungsfaktor und nur einen für den maximalen verwenden ... Sie haben die Wahl.

Mit nur diesen beiden **Attribute nodes** erhalten wir jedoch eine gewisse Vielfalt aus einem Objekt, das aus einer einzigen Geometrie besteht: **Stone 1**.

Wenn wir den Typ des zu instanziiierenden Objekts in **Point Instance** von **Object** auf **Collection** ändern und eine Sammlung von Kieselsteinen angeben, werden die Dinge viel interessanter!

Wählen wir jedoch erstmal **Object** mit **Stone 1** in **Point Instance** aus, um eine andere Funktionalität zu untersuchen, die mit **Geometry Nodes** verfügbar ist.

In klassischen Partikelsystemen können wir auch definieren, auf welchen Teilen der Geometrie die Partikel erzeugt werden sollen und auf welchen nicht; dies geschieht insbesondere mit **Weight Painting**, indem wir Dichtewerte (von 0 bis 1) auf den verschiedenen Teilen einer Oberfläche definieren ... nun, wir können es auch hier tun!

Bevor wir jedoch fortfahren, müssen wir die Anzahl der Unterteilungen der **Plane** erhöhen, denn das **Weight Painting** (das außerhalb des **Geometry Node** auf die ursprüngliche Geometrie angewendet werden muss) wirkt auf die Scheitelpunkte, so dass die 4 Basisscheitelpunkte der Ebene nicht ausreichen.

In einer 3D-Ansicht gehen wir also in den **Edit Mode** und unterteilen die gesamte Fläche mehrmals; dann schalten wir in den **Weight Paint Mode**.

Standardmäßig sollte die Oberfläche blau erscheinen, d.h. "Gewicht 0"; der **Point Distribute node** erzeugt die Punkte auf den Flächen mit Gewicht 1, wir müssen also die Flächen einzeichnen, auf denen wir die Instanzen erscheinen lassen wollen.

HINWEIS: wenn du die Instanzen überall anwenden willst, außer an bestimmten Stellen, musst du umgekehrt vorgehen; in diesem Fall benutzen wir den Pinsel in den Bereichen, die wir von **Point Distribute** ausschließen wollen (und **Blender** wird automatisch eine **Vertex Group** erstellen, die "Group" als Standardnamen haben sollte, aber wir können sie auch in etwas Aussagekräftigeres umbenennen, wie z.B. "**Pebbles**").

Am Ende der Operation öffnen Sie das Menü "**Weights**" im 3D-Fenster und klicken Sie auf "**Invert**", um die Gewichte umzukehren: jetzt wird der Bereich, den wir gemalt haben, das Gewicht 0 haben, während wir im Rest der Geometrie das Gewicht 1 haben werden.

Natürlich hindert uns nichts daran, die Gewichte zu einem späteren Zeitpunkt zu ändern, indem wir den Wert des Parameters "**Weight**" auf der Registerkarte "**Tool**" im Modus "**Weight Paint mode**" ändern und die Oberfläche bürsten; wir müssen uns nur daran erinnern, dass wir KEINE Instanzen in Bereiche mit 0-Gewichtung einfügen werden.

Im Editor **Geometry Nodes** geben wir den Namen der **Vertex Group** in das Feld **Density** des **Point Distribute node** ein ("**Pebbles**", in unserem Fall; achten Sie auf Groß- und Kleinschreibung): die Verteilung der Punkte auf der ursprünglichen Oberfläche wird proportional zu den von uns festgelegten **Vertex Weights** sein.

Bevor wir dieses Videotutorial und damit die Miniserie EINFÜHRUNG in **Geometry Nodes** in **Blender 2.92** abschließen, wollen wir uns noch ein weiteres Beispiel für die Instanziierung von Objekten ansehen ... allerdings mit einer weiteren Änderung.

Innerhalb des Schemas, das wir definiert haben, fügen wir einen neuen **Point Density Node** ein und verbinden ihn mit dem **Geometry**-Port von **Group Input**, dann verbinden wir ihn mit einer neuen (leeren) **Point Instance** und verbinden die beiden in der Szene vorhandenen **Point Instances** mit einem neuen **Join Geometry Node**, der wiederum mit dem bereits in der Szene vorhandenen **Join node** verbunden wird; in der Praxis wird die resultierende Geometrie durch die ursprüngliche gegeben sein, die mit der durch zwei **Instance nodes** hinzukommenden verbunden wird.

Im neuen **Distribute node** setzen wir das gleiche **Density**-Attribut "**Stone 1**", aber in **Point Instance** setzen wir das **High Grass**-Objekt als das zu instanziiierende.

Falls erforderlich, können wir einen **Point Rotate Node** einfügen, um die Grashalme korrekt auszurichten.

Wir fügen auch einen **Point Scale Node** ein (zwischen **Point Rotate** und **Point Instance**) und stellen insbesondere den Modus **Vector** ein, so dass wir den Skalierungsfaktor der Grasgröße mit einem Vektor angeben können.

Fügen wir einen **Combine XYZ Node** ein und verbinden seinen **Vector**-Ausgang mit dem **Factor** (Vektor) Eingang von **Point Scale**; die Grashalme werden verschwinden, weil die Parameter **Combine XYZ** einen Standardwert von 0,0 haben werden, also setzen wir XY auf 1,0 zurück (oder auf jeden Fall auf einen angemessenen Wert für diese Dimensionen).

Anschließend verbinden wir den **Z**-Eingang von **Combine XYZ** mit einem neuen **Geometry Input**. Wie Sie vielleicht schon erraten haben, soll damit die Höhe der Grashalme in der Modifizierungsschnittstelle angepasst werden ... aber warum sollten wir uns auf einen konstanten Wert beschränken oder diesen Wert manuell ändern müssen?

Fügen wir dem **Blender**-Fenster einen **Timeline**-Editor hinzu und legen einen Frame-Bereich für eine Animation fest, zum Beispiel **240** Frames.

Gehen wir zu Frame 1, setzen wir den **Z**-Parameter auf 0,0 (den wir nach Belieben umbenennen können, wie in der vorangegangenen Episode mit den Namen der Uhrzeiger); dann klicken wir mit der rechten Maustaste auf den Wert dieses Parameters und wählen **"Insert Keyframe"**, um einen Keyframe der Animation aufzunehmen.

Gehen wir zu Frame 240 der Animation, ändern wir den Wert von **Z** wie gewünscht und fügen wir auch für diesen Parameter einen Keyframe am letzten Frame unserer Animation ein.

Jetzt müssen wir nur noch auf **Play** in der **Timeline** drücken und das Ergebnis unserer Animation betrachten.

So, das war's auch schon mit diesem Tutorial!

Wir sehen uns bald wieder!