

# Einführung in Geometry Nodes in Blender

## 2.92 - Tutorial 1 von 2



Das in diesem Tutorial verwendete 3D-Modell: Office wall clock 1  
(<https://sketchfab.com/3d-models/office-wall-clock-1-7fafd736d520456eac2066808b0a0006>)  
(<https://sketchfab.com/3d-models/office-wall-clock-1-7fafd736d520456eac2066808b0a0006>)

Hallo zusammen!

Dies ist das erste von zwei Einführungs-Tutorials zu **Geometry Nodes** in **Blender** (erstellt insbesondere mit der Version 2.92 der Software). Diese Tutorials sind für diejenigen gedacht, die bereits über Grundkenntnisse in **Blender** verfügen, denn ich werde einige Informationen über die Benutzeroberfläche und die Modellierung (sowohl im **Object-** als auch im **Edit Mode**) voraussetzen.

**Geometry Nodes** ist ein Werkzeug, das mit der **Blender**-Version 2.92 eingeführt wurde und das es ermöglicht, Objekte auf parametrische Weise zu transformieren, d.h. mit Hilfe von Werten.

Auf Youtube gibt es viele Tutorials, die vor allem zeigen, wie man **Geometry Nodes** verwendet, um ein Partikelsystem zu ersetzen, um Objektinstanzen auf (oder in) andere zu "streuen", wie z. B. Felsen und Grashalme auf einer Oberfläche ... nun, im nächsten Tutorial werde ich das gleiche Beispiel auch hernehmen, aber für dieses erste Tutorial zu diesem Thema möchte ich mich auf einige grundlegende Konzepte konzentrieren.

Wir haben gesagt, dass **Geometry Nodes** ein Modifikator ist, und wir wissen, dass sich Modifikatoren von **Blender**-Operatoren unterscheiden, weil sie es erlauben, Transformationen auf nicht-destruktive Weise durchzuführen, was bedeutet, dass es möglich ist, einige Parameter auch zu einem späteren Zeitpunkt zu ändern und die Änderungen auch in den anderen Modifikatoren angewandt zu finden; zum Beispiel zeige ich im Video den Unterschied zwischen einer Kolonnade, die mit dem **Array modifier** (oben) und einer

anderen, die mit dem **Spin-Operator** (unten) gemacht wurde: Um die Höhe der Säulen zu ändern, muss ich im Fall von **Array** nur in den **Edit Mode** gehen und die Grundgeometrie anheben, um die Änderung in allen anderen Elementen zu finden; mit dem **Spin Operator** hingegen muss ich alle erstellten Elemente löschen, das erste Element im **Edit Mode** ändern und dann erneut **Spin** anwenden.

Außerdem können wir, sobald wir eine Operation mit einem **Operator** durchgeführt haben, die angewendeten Parameter nicht mehr ändern, wenn wir zu einem anderen Werkzeug wechseln, während bei den **modifiers** diese Parameter weiterhin verfügbar sind und wir sie auch zu einem späteren Zeitpunkt ändern können (bis wir den **modifier** anwenden, natürlich).

Dies ist ein sehr elementares Beispiel, aber der Vorteil der Verwendung von **modifiers** - wenn möglich - liegt auf der Hand, insbesondere wenn die Gefahr besteht, verschiedene Änderungen an den Elementen vornehmen zu müssen oder, schlimmer noch, Operationen nacheinander durchführen zu müssen, bei denen eine Änderung uns dazu zwingen kann, auch die nachfolgenden Operationen erneut durchführen zu müssen; das Gesagte erklärt auch, warum die Modellierung mit **modifiers** als parametrisch und nicht-destruktiv bezeichnet wird.

**Geometry Nodes** ist also ein **modifier**, aber im Gegensatz zu den anderen grundlegenden **modifiers** von **Blender** kann er durch die Kombination von **Nodes** angepasst werden.

**Nodes** sind im Grunde die visuelle Darstellung von zwei Arten von Elementen: Informationen oder Operationen; wenn Sie Erfahrung mit der Erstellung von Materialien oder der Vorproduktion in **Blender** haben, wissen Sie zum Beispiel, dass Sie eine Eingangsinformation nehmen können (definiert in einem **Node**), Operationen durchführen können (durch die Kombination von einem oder mehreren Zwischen-**Nodes**) und am Ende das Ergebnis als Ausgabe erhalten (die wiederum in einem **Nodes** definiert wird). Sehen wir uns also an, wie dieses Konzept auf die Transformation von Objekten unter Verwendung von **Geometry Nodes** angewendet wird.

Als Modell für das Beispiel dieses Tutorials verwende ich eine Wanduhr; du kannst den Link zu diesem Modell in der Beschreibung des Videos finden, aber es ist nicht notwendig, es zu haben: du kannst dem Tutorial folgen und die Konzepte lernen, ohne sie notwendigerweise mit diesem Modell zu replizieren!

Ich habe die **Blender**-Oberfläche für die Arbeit mit **Geometry Nodes** eingerichtet, indem ich einen Editor vom Typ "**Geometry Node**" unter der 3D-Ansicht und die Registerkarte "**Modifiers**" für das ausgewählte Objekt im "**Properties**"-Editor geöffnet habe (Hinweis: Ich habe dieses Layout hier eingestellt, weil ich das Video auf einem Monitor aufnehme; wenn Sie einen zweiten Monitor verwenden, empfehle ich Ihnen, den **3D-Viewpoint** und den Editor "**Properties**" auf einem Monitor zu verwenden und den Editor "**Geometry Nodes**" auf dem anderen zu öffnen).

Der Minuten- und der Stundenzeiger sind separate Objekte, die der Uhr übergeordnet sind; beide Objekte haben den Ursprung dort, wo er sein sollte (in ihrem Drehpunkt), so dass eine Drehung um die **Y**-Achse (lokale Achse) jedes Objekts zu einer korrekten Drehung führt. Für dieses Tutorial stellen wir die **Origin** eines der beiden Zeiger wieder her (indem wir ihn auswählen und **Cursor to selected** wählen, aus dem Menü, das wir mit **SHIFT S** öffnen können), dann löschen wir die drei in der Szene vorhandenen Zeiger.

Dann fügen wir einen **Cube** hinzu, dessen **Origin** an der Stelle liegt, an der sich die anderen Hände befanden. Das ist wichtig, weil die Transformationen um diesen Punkt herum stattfinden wird.

Der **Cube** wird an der Basis der ursprünglichen Uhr angebracht, so dass wir den **Cube** bei der Transformation der Uhr ebenfalls verschieben, drehen und in der Größe verändern können.

Fügen wir einen **Modifier** für **Geometry Nodes** hinzu; dieser Vorgang kann über das Menü **Modifier** des Objekts oder über die Schaltfläche "**New**" im Editor für **Geometry Nodes** ausgeführt werden.

Standardmäßig werden zwei **Nodes** erstellt, die "**Group Input**" und "**Group Output**" genannt werden; die Eingabe liefert die Geometrie des Objekts, wie sie dem **Modifier** zugeführt wird (in diesem Fall wurde der Zeiger bereits modelliert und die **Origin** manuell festgelegt): Der **Geometry Input** beginnt mit diesen Daten), während die Ausgabe das ist, was am Ende der gesamten Verarbeitung zurückgegeben wird, die wir dann zwischen diesen beiden **Nodes** durchführen werden.

Die **Nodes** werden hinzugefügt, indem man sie aus dem Menü **Add** (Tastenkürzel: **SHIFT A**) auswählt; in diesem Menü ist besonders zu beachten, dass die **Nodes** in verschiedene Gruppen unterteilt sind, so wirken zum Beispiel die **Nodes** der Gruppen **Mesh** und **Geometry** auf das gesamte Objekt und seine Geometrie (mit Operationen im **Object-** und **Edit Mode**), während die der Gruppe **Point** auf die erzeugten Partikel wirken; andere Knoten dienen zum Beispiel dazu, Werte zu liefern, mit denen die Objekte transformiert werden (Gruppe **Input**), während andere dazu dienen, diese Werte zu verändern (Gruppen **Utilities** und **Vector**).

Um den Würfel in den Minutenzeiger zu transformieren, fügen wir einen **Geometry - Transform node** hinzu und platzieren ihn genau in der Mitte der Verbindung, die von der **Group Input** zur **Group Output** führt (**Blender** stellt die Verbindungen automatisch her).

Werfen wir einen Blick auf die Farbe der Eingangs- und Ausgangsports der verschiedenen **Nodes**, mit denen wir den Datenfluss und ihre Operationen steuern: Die **Geometries** haben eine grüne Farbe und zeigen an, dass Informationen über die gesamte Geometrie des Objekts dorthin weitergeleitet werden; die Gates vom Typ **Vector** (die Triplets von numerischen Werten enthalten, einen für jede Rotationsachse) sind dagegen lila.

Wir arbeiten mit **Translation** und **Scale**, um den Minutenzeiger zu definieren. Insbesondere werden wir **Translation Z** auf 0,049 und **Scale XYZ** auf 0,001, 0,001 bzw. 0,050 setzen.

Ich möchte sofort auf eine Sache hinweisen: Die Drehung um die lokale **Y**-Achse, um die Minuten zu definieren, sollte **NICHT** in diesem **Node** durchgeführt werden, da hier die Operationen in Bezug auf die verschobene und in der Größe veränderte Geometrie stattfinden.

Um um die **Origin** des Würfels zu rotieren, müssen wir einen neuen **Geometry - Transform node** zwischen den soeben erstellten **Nodes** und der **Group Output** hinzufügen: Dort können wir um die **Y**-Achse rotieren.

Eine weitere Besonderheit ist, dass diese Transformationen sozusagen "innerhalb des Modifikators" stattfinden: Wenn wir die Registerkarte **Item - Transform** in der 3D-Ansicht betrachten, stellen wir fest, dass in Wirklichkeit **Location**, **Rotation** und **Scale** nicht auf das ursprüngliche Objekt angewendet wurden. Wenn wir außerdem in der 3D-Ansicht in den **Edit Mode** wechseln, stellen wir fest, dass der Würfel in Wirklichkeit immer noch da ist, mit seinen ursprünglichen Eigenschaften!

Dieses Verhalten, das in jeder Hinsicht dem von **Subdivision Surface** ähnelt, lässt uns verstehen, dass **Geometry Nodes** in der Tat ein **Modifier** ist (und in der Tat ist es möglich, ihn auf der Registerkarte **Modifier** anzuwenden, wodurch die Transformationen endgültig werden ... aber wenden Sie ihn jetzt noch nicht an).

Gehen wir zurück zum **Geometry Node**, wählen wir den ersten **Transform node** aus und duplizieren ihn (mit **SHIFT D**), dann ziehen wir den **Geometry**-Ausgang von **Group Input** auf den **Geometry**-Eingang dieses neuen **Node**, dann übergeben wir den Ausgang dieses **Node** an den **Geometry Input** von **Group Output**.

Innerhalb des **Node** ändern wir **Scale Z** auf 0,025 und **Translation Z** auf 0,025 und erhalten so einen Stundenzeiger.

Wie wir sehen, ist es möglich, die ursprüngliche Geometrie des Objekts (in diesem Fall der Würfel) zu nehmen und sie von **Group Input** an mehrere Operatoren weiterzugeben ... aber wie erhält man BEIDE geänderten Geometrien im selben **Object**?

Ganz einfach: durch Verwendung eines **Geometry - Join Geometry node**, der es uns ermöglicht, mehrere Geometrien zu einer einzigen zu verbinden und das Ergebnis an andere **Nodes** weiterzugeben; das ist also das, was der **Join operator** im **Edit Mode** tut, aber hier verwenden wir ihn auf nicht-destruktive Weise, innerhalb eines **Modifier**!

Wir setzen den Wert **Rotation Y** des Minutenzeigers auf 0 zurück, duplizieren diesen Knoten und platzieren die Kopie in den Arbeitsablauf des soeben erstellten Stundenzeigers: Auf diese Weise können wir den Rotationswinkel sowohl für die Stunden als auch für die Minuten unabhängig voneinander einstellen.

Schauen wir uns also an, wie man eine kleine Schnittstelle mit **Geometry Nodes** erstellen kann, um schnell Stunden und Minuten in der Registerkarte **Modifier** einzustellen und die Zeiger automatisch zu bewegen.

Wir wollen die **Rotation Y** auf der Grundlage der Zeit einstellen (eine Zahl zwischen 0 und 60 für Minuten und zwischen 0 und 12 für Stunden. Einschließlich 60 und 12). Beginnen wir mit der **Rotation** der Minuten.

Beachten Sie, dass **Rotation** einen violetten Einstiegsanschluss hat, es ist also ein Vektor. Um die Eingangskanäle zu trennen, fügen wir einen **Vector - Combine XYZ node** hinzu und verbinden den **Vektor**-Ausgang des soeben hinzugefügten Knotens mit dem **Rotation**-Eingang von **Transform**, dann verbinden wir den **Y**-Eingang von **Combine XYZ** mit einem neuen Eingang von Gruppeneingang (ziehen Sie einfach die Verknüpfung von **Combine XYZ** auf den halb sichtbaren Eingangsanschluss von **Group Input** unter **Geometry**).

Auf der Registerkarte **Geometry Nodes modifier** wird der Eingabeparameter **"Value"** mit einem Standardwert angezeigt. Um den Namen dieses Parameters in **"Minutes"** zu ändern, öffnen Sie die **Sidebar** des Editors **"Geometry Nodes"**

(Tastaturkürzel: **N**, während sich der Mauszeiger in diesem Editor befindet), und ändern Sie den Namen von "**Value**" in "**Minutes**".

Wir setzen auch den Wert **0.0** für **Min** (kleinster Wert, den der Benutzer eingeben kann) und **60.0** für **Max**.

Bei der Eingabe eines beliebigen Wertes zwischen 0 und 60 stellen wir jedoch fest, dass sich der Minutenzeiger nicht angemessen dreht; dies liegt daran, dass der Wert des Winkels **Y** in Radiant ausgedrückt wird und insbesondere der runde Winkel den Wert 6,28 hat (d. h. das Doppelte von Pi).

Um diese Zuordnung vorzunehmen (d. h. einen Wert aus dem Bereich 0 - 60 in den Bereich 0 - 6,28 zu bringen), fügen wir einen **Utilities - Map Range node** zwischen dem Ausgang "**Minutes**" von **Group Input** und dem **Y**-Eingang von **Combine XYZ** ein und setzen dann die Werte des **Node** (die eine ziemlich intuitive Bedeutung haben) folgendermaßen:

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

Jetzt können wir auf der Registerkarte **Modifier** einfach den Wert der Minuten eingeben, die auf der Uhr erscheinen sollen, und der Minutenzeiger dreht sich gleichmäßig: Das ist bequemer, als das Objekt um die **Y**-Achse drehen zu müssen!

Wie beim Stundenzeiger ist das Muster der **Nodes** praktisch identisch, mit Ausnahme der Zuordnungswerte im Knoten **Map Range**; gehen wir also folgendermaßen vor:

- Wir wählen und duplizieren **Map Range** und **Combine XYZ** des Minutenzeigers und platzieren die Kopien in einem leeren Bereich des Editors;

- Wir verbinden den **Vector** des neuen **Combine XYZ** mit dem **Rotation-**Eingang des Stundenzegers;
- Wir verbinden den **Value** des neuen **Map Range** mit einer neuen **Group Input** Ausgang (den wir unter anderem in **Hours** umbenennen können);
- Ändern wir den **"From Max"**-Wert des **Map Range Node** der **Hours** von 60 auf 12.

ACHTUNG: Wählen Sie die **Group Input** aus und stellen Sie sicher, dass die **Max** Werte der Eingabeports für **Minutes** und **Hours** korrekt auf 60 bzw. 12 eingestellt sind!

Jetzt können wir die Stunden und Minuten der Zeiger in der **Blender**-Oberfläche einstellen und die gewünschte Zeit angeben ... sind wir fertig? Nun, nein: Bei einer echten Uhr bewegt sich der Stundenzeiger mit dem Minutenzeiger vorwärts (von einer Stunde zur nächsten), während dies bei diesem Modell nicht der Fall ist (versuchen Sie zum Beispiel, "2:50" oder "4:55" einzustellen: der Stundenzeiger bleibt bei 2 oder 4 stehen, während er nahe bei 3 oder 5 sein sollte).

Was wir brauchen, ist ein kleines Inkrement, ein "Offset" der Drehung des Stundenzegers, das insbesondere höchstens 30 Grad betragen darf (ein Zwölftel eines Drehwinkels, wenn der Minutenzeiger nahe bei 60 steht).

Zuerst müssen wir also den Wert des **Minutes**-Eingangs nehmen und ihn durch 2 teilen, was wir mit einem **Utilities - Math node** tun können, indem wir seinen Modus auf **Divide** ändern und **"Minutes"** auf den ersten Wert und 2 auf den zweiten Eingangswert setzen.

Das Ergebnis ist ein Rotations-Offset, der dem Stundenzeiger zugeführt wird, aber dieser Wert ist in Grad ausgedrückt, während wir ihn in Radianten haben möchten ... nun, wir können ihn leicht durch einen weiteren **Utilities - Math node** umwandeln, dieses Mal in **"To Radians"**!



Um diesen Wert zu dem aus dem **Map Range** des Stundenzeigers stammenden Wert (der bereits im Bogenmaß angegeben ist) hinzuzufügen, fügen wir einen weiteren **Utilities - Math node** hinzu, diesmal im **Add** Modus, und fügen **Map Range** und die Ausgabe von **"To Radians"** hinzu, um dann das Ergebnis der Operation an **Combine XYZ** und schließlich an die **Y-Drehung** des Stundenzeigers zu übergeben.

Theoretisch funktioniert an dieser Stelle alles, wie wir durch einige Tests mit der Schnittstelle in der Registerkarte **Modifier** sehen können; außerdem, wenn wir die Uhr transformieren, werden die Änderungen auch konstant auf die Zeiger angewendet ... oder besser: sie werden auf das **Cube**-Objekt angewendet!

Die beiden Zeiger sind in Wirklichkeit Geometrien desselben ursprünglichen **Cube**, der immer noch vorhanden ist und (als Kind) mit der Uhr verbunden ist: alle Änderungen wurden im **Geometry Nodes node** vorgenommen und können später geändert werden, bis wir den **Modifier** anwenden.

Bevor wir dieses Tutorial abschließen, noch eine letzte Bemerkung: Die Felder **"Hours"** und **"Minutes"** auf der Registerkarte **"Modifier"** akzeptieren auch dezimale Werte und berücksichtigen sie bei ihren Berechnungen, so dass das visuelle Ergebnis falsch sein kann (versuchen Sie zum Beispiel, 0 für Minuten, aber 9.30 für Stunden einzugeben).

Eine unelegante, aber funktionelle Lösung für dieses Problem ist das Einfügen des **Utilities - Math nodes** im Modus **Rounding - Floor** (d. h. Runden auf die nächstniedrigere ganze Zahl) unmittelbar nach den Anschlüssen **"Minutes"** und **"Hours"** der **Group Input**; dadurch werden die Dezimalzahlen in der Modifikatorschnittstelle zwar nicht ausgeblendet, aber zumindest bei den Berechnungen ignoriert.

Hier, jetzt sind wir wirklich fertig mit dem Interface unserer Uhr ... und auch mit diesem ersten Tutorial über **Geometry Nodes** in **Blender 2.92**! Im nächsten (und letzten für diese Miniserie) werden wir uns ein Beispiel für die Streuung

von Objekten über eine Fläche ansehen.

Bis bald!