

2024

ISSN 1433-2620 > 28. Jahrgang >> www.digitalproduction.com

Publiziert von BG Germany GmbH

Deutschland € 17,90

Österreich € 19,-

Schweiz sfr 23,-

3

DIGITAL
PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

MAI | JUNI 03:2024



Schönes!

The Nevers, Fritzi & Sophie, Planet der Affen

Schlaues!

Comfy UI, Look Development, Flame

Starkes!

Unreal 5.2, SynthEyes, Obsbot, Twixtor, Prism

Seltsames!

35 mm in Virtual Production, Plug-ins und KIs

Procedural Worldbuilding mit Unreal

Wir bauen Realtime Environments mit Unreals Procedural Content Generation Framework (PCG) – oder wollt ihr wirklich alles einzeln von Hand machen?

von Cornel Hillmann

Als Epic Games mit Unreal 5.2 die erste experimentelle Version des PCG Frameworks vorstellte, war die Aufregung unter UE5 Usern zunächst groß – prozedurale Tools gab es schon vorher in der Unreal Welt, diese waren auf Blueprint oder C++ Plug-in Basis gebaut, in den Möglichkeiten kommen, diese natürlich nicht an eine natives PCG Framework oder eine Houdini Integration heran. Aufwendige prozedurale Environments zu generieren war bis dato nur über eine Houdini-Integration möglich. Bekannte frühe Beispiele dafür sind Titel wie Far Cry 5 oder auch Spiderman (PS4), wo über Houdini Connect mit direkter Engine Anbindung gigantische Maps durch SideFXs prozedural Tools konstruiert wurden.

Die erste Frage, die dann sofort von aufstrebenden Technical Artists diskutiert wurde, war: Wird Unreal PCG Houdini kurz oder mittelfristig ersetzen? Lohnt sich das Zeitinvestment in diese Tools eventuell nicht mehr?

Houdini, Blender, oder Unreal?

Das ist natürlich nicht der Fall, denn Houdini wird auch sicherlich in der Zukunft ein fester Bestandteil in den Pipelines bei großen Entwicklern wie Ubisoft bleiben. Dafür ist die hochkomplexe, aber äußerst effektive Kom-

ination von Houdini Digital Assets (HDAs), VEX Scripting und Session Sync viel zu weit seiner Zeit voraus. Aber es zeigt auf der anderen Seite, dass prozedurale Tools überall auf dem Vormarsch sind, wie man es ja auch bei Blenders Geometry Nodes sieht. Die Vorteile liegen auf der Hand: Wesentlich schneller komplexe Szenen zu gestalten, während man gleichzeitig totale kreative Kontrolle über das Resultat hat.

Die zweite Frage, die in diesem Zusammenhang öfter einmal in Foren diskutiert wurde, war dann: Welche Rolle spielt der Environment Artist hier eigentlich noch? Oder, wer ist hier am Drücker: Ist das noch Biomasse oder schon KI? Diese Frage ist natürlich nachvollziehbar, insbesondere vor dem Hintergrund, dass mancher schon die Trompeten der KI-Apokalypse tönen hört.

Aber auch hier kann man natürlich vorerst Entwarnung geben, denn zwar wird über PCG Algorithmen dynamischer Content generiert, dennoch unter voller Kontrolle und dem Gestaltungswillen menschlicher Gehirnzellen.

Die Unreal PCG Era beginnt

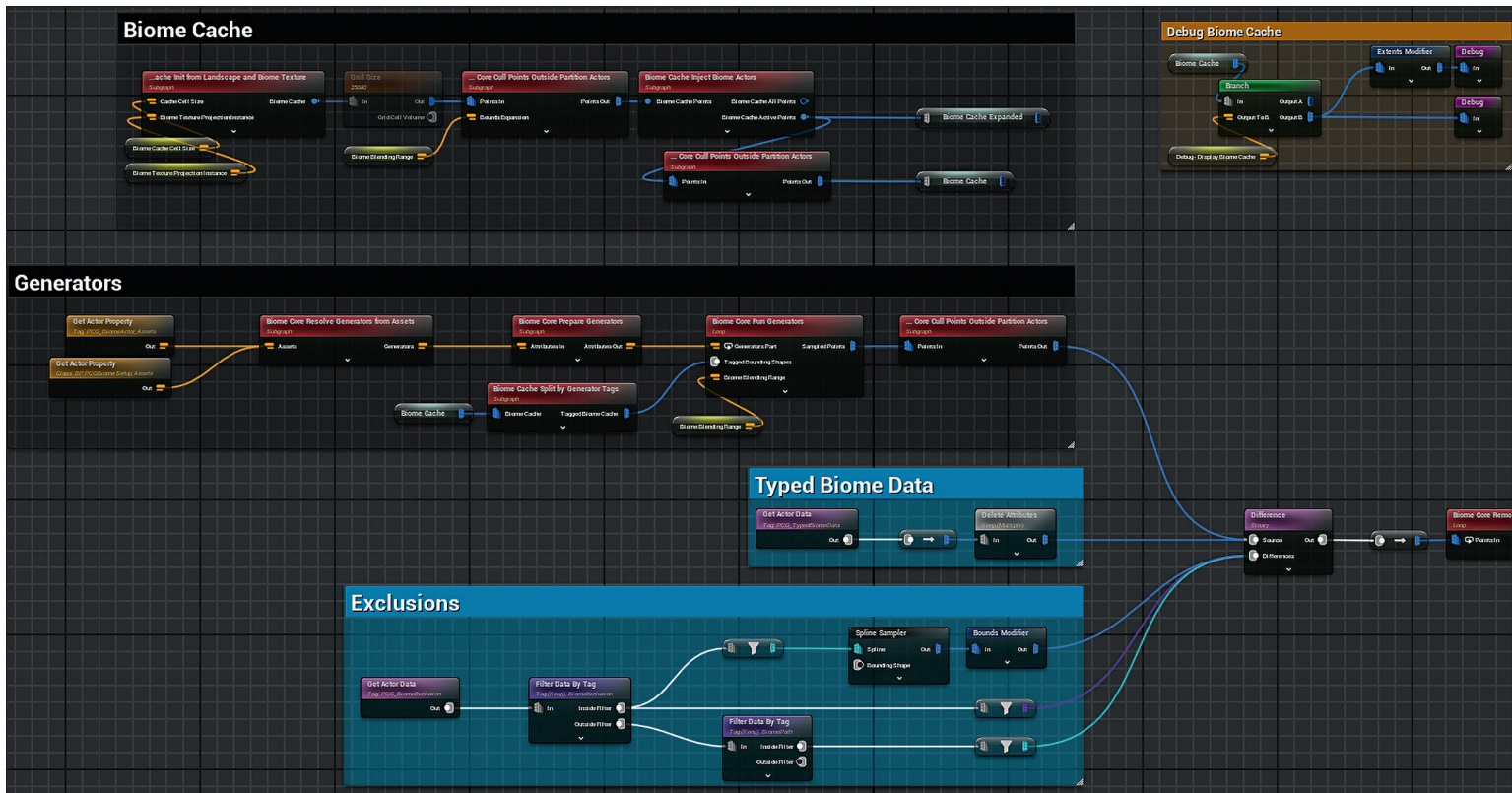
Unreals PCG Framework ist in der Benutzung ähnlich dem bewährten Blueprint Visual Scripting System. In dem Node basierten





In Epic Games
Electric Dreams PCG
Demo navigiert man
durch einen endlos
wirkenden Urwald,
der durch detailtiefe
und endlose Varia-
tionen immer
lebendig wirkt.
Über PCG Parameter
lassen sich Dichte
und Szenen-Detail
beliebig steuern
und verändern.





Mit dem neuen PCG Biome Core- und Biome Sample-Plug-in aus der Unreal 5.4 Preview Version zeigt Epic Games anhand der neuen Nodes, wie die standardisierte prozedurale Landschafts-Generation der Zukunft aussehen soll.

Umfeld werden räumliche Daten abgegriffen, daraus Punkte mit PCG Eigenschaften generiert, die wiederum neu sortiert, gefiltert und im Datenfluss verarbeitet werden, um dann aus den Konditionen der PCG-Logik die Endposition der Punkte zu errechnen, an denen das Mesh aus seinen entsprechenden Parametern generiert wird.

Der spezielle PCG Graph ist auch denkbar geschmeidig in das Unreal Ökosystem eingebettet, er lässt sich als Komponente in Blueprint Actors erweitern und bleibt dank Nanite's Virtualized Micropolygon Geometry System skalierbar und performant.

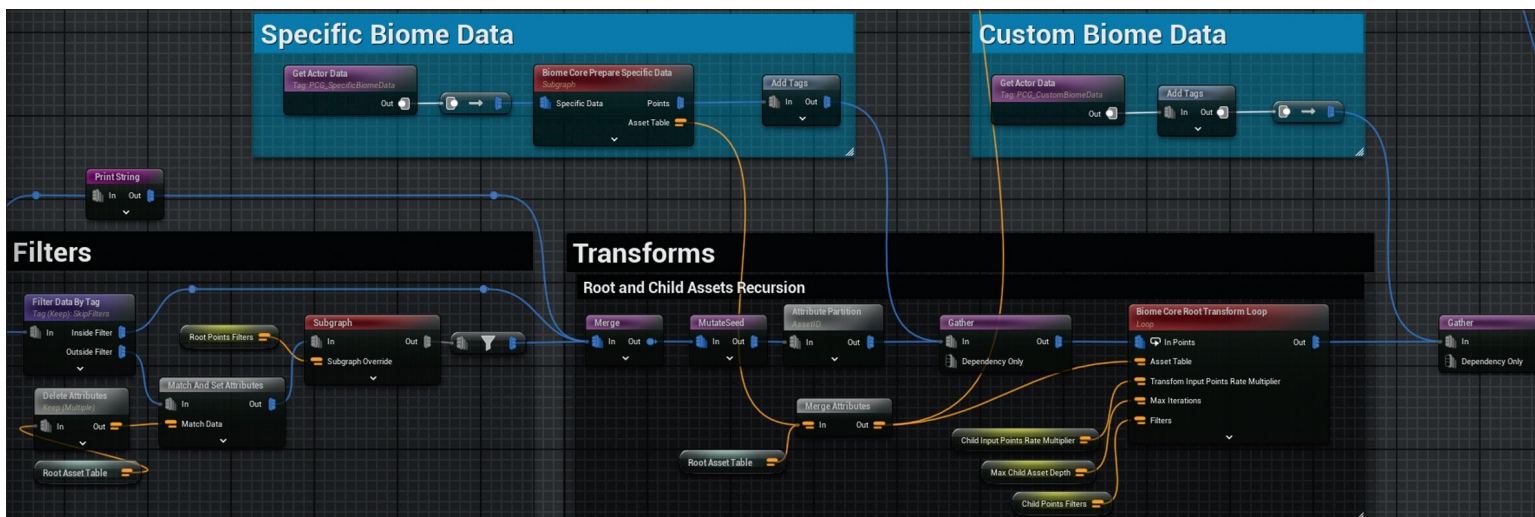
Erstaunlicherweise funktioniert PCG auch „At Runtime“, es lassen sich also in Realtime PCG Graphen ausführen, was sprichwörtlich Türen zu neuen Welten aufstößt. Prozedurale Welten, wie man es vom Prinzip her schon aus Titeln wie etwa No Man's Sky kennt, können natürlich auch in der Virtual Produktion nützlich sein, um zum Beispiel in Echtzeit Szenen-Charakteristik über User Parameter zu steuern.

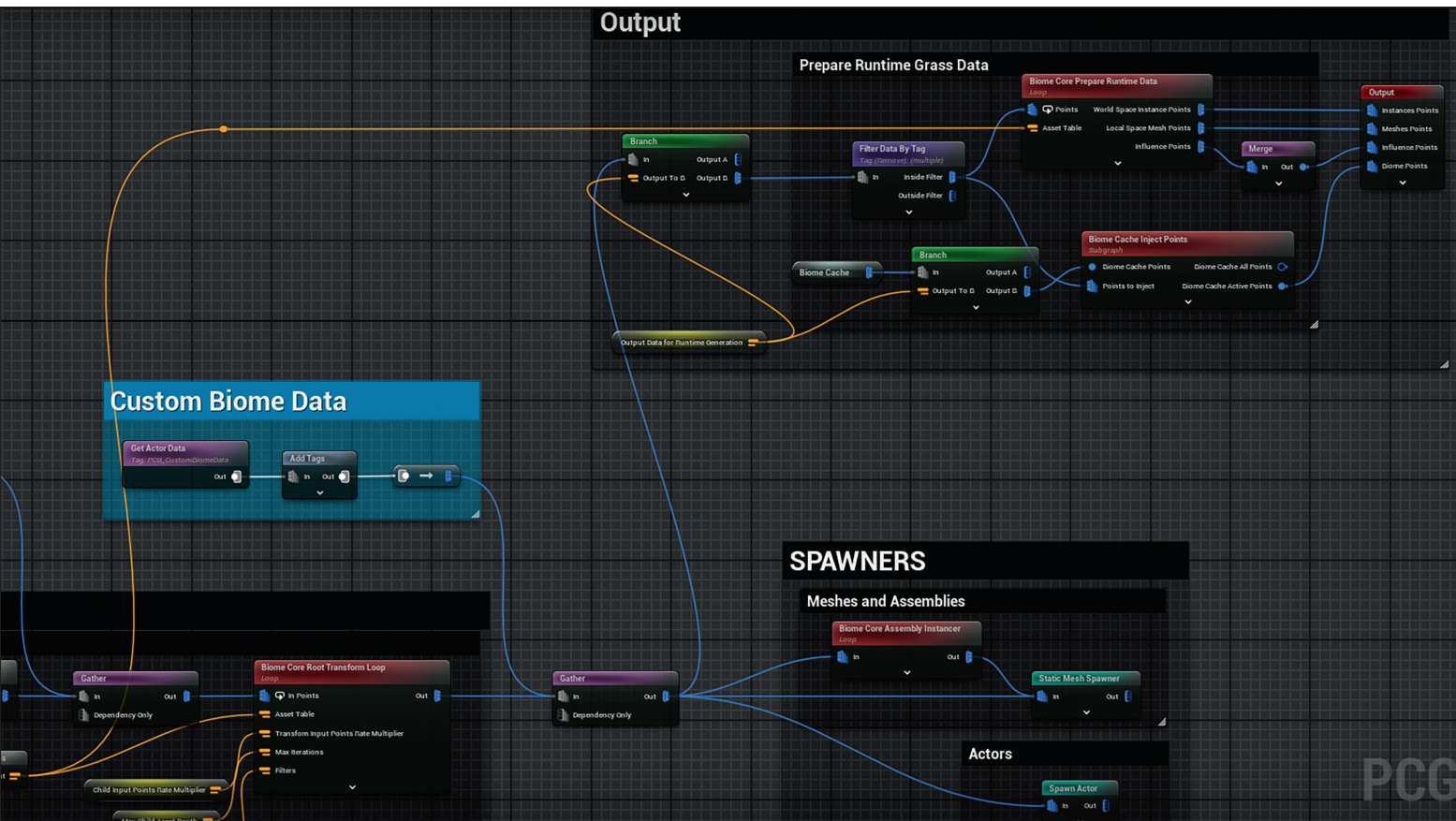
In den bisherigen Anwendungen und Beispielen von Epic Games ist der Fokus auf Landschaft und Natur gelegt, das komplexe Thema Gebäude, Infrastruktur und Städte

dagegen wird eher verhalten angegangen. Es wird zur Zeit noch daran gearbeitet, diese Bereiche benutzerfreundlicher zu machen, ließ man bei den Developer Streams im Q&A durchblicken.

UE 5.4: Prozedural in die Zukunft

Nachdem Epic Games mit Unreal 5.2 das Kapitel PCG aufgeschlagen hat, wurde in der Version UE 5.3 einiges im Detail vereinfacht. Mit der Version UE 5.4 werden nun jedoch einige entscheidende Performance-Verbes-





Biome Core Plug-in: Ein Showcase für die neuen PCG Features und Nodes

serungen und jede Menge neuer Nodes nachgereicht, die den Workflow wesentlich erleichtern.

So gibt es nun eine Water Spline Interop Node, die es wesentlich einfacher macht, Tiefe und Weite entlang von Water Splines zu sampeln. Gerade um Flussdaten (River Water Body) zu erfassen, waren vorher mathematische Verrenkungen nötig gewesen, um brauchbare Ergebnisse zu bekommen. Eine echte kleine Überraschung in der UE 5.4 Preview ist der neue Biome Core Plug-in. Einmal aktiviert, demonstriert Epic hier anhand einer Beispiel Level Map, wie die Zukunft der standardisierten Biome Erstellung aussehen soll.

Zum einen ist dies auch ein Showcase für die neuen hierarchischen Logik Funktionen wie Recursive Subgraphs, zum anderen eine Initiative für eine Art Standard PCG Graph Template für Naturelemente, das jeweils der Situation angepasst werden kann, ohne immer bei Null anfangen zu müssen. Auf der GDC 2024, State of Unreal Präsentation ([is.gd/unreal_gdc_2024](https://www.gdcvault.com/play/1024024)) wurden dazu auch einige praktische Beispiele gezeigt. Insbesondere Lego, neben Disney, einer der größten Partner von Epic, zeigte hier die flexiblen Einsatzmethoden für PCG Landschaften in der bekannten Bauklötzoptik.

Unterstützt werden die neuen 5.4 PCG-Features mit weiteren Neuerungen, die für das Environment Design wichtig sind. Nanite

Dynamic Tessellation, erlaubt dynamisches Landscape-Displacement at Runtime, also in Echtzeit, um zum Beispiel Oberflächen-Details zu ändern. Die Funktion Nanite Spline Meshes, nützlich für zum Beispiel Straßen-Generierung, wurde optimiert und ist jetzt „Production Ready“. Für heterogene Volummen, also realistische Feuer und Rauch Simulation, gibt es nun bessere Render-Integration mit Schatten und transparenten Objekten. Also viele Verbesserungen im Detail was Landschafts-Themen betrifft, neben den großen Show-Favoriten, wie dem neuen Motion-Graphics-Mode und den neuen Animations-Tools: Motion Matching, automatisches Retargeting und den tollen neuen Control-Rig Modulen.

Mit Unreal Version 5.4 wurde ein beeindruckendes Update vorgelegt, gerade was Performance betrifft. Der Weg ist damit frei, um riesige und gleichzeitig detailverliebte Realtime-Umgebungen zu bauen, sei es für Open World Games oder als Environments für die virtuelle Produktion.

Finster ruft es aus dem Wald: Die Electric Dreams Demo

Um die neuen prozeduralen Tools in Unreal nutzen zu können, muss zunächst das PCG Plug-in aktiviert werden. Einen schnellen Überblick über die Möglichkeiten verschafft man sich am einfachsten mit dem Electric

Dreams Beispiel Projekt von Epic Games, das Mitte 2023 vorgestellt wurde. Mit diesem eindrucksvollen Beispiel, das durchaus einige Hardware Ansprüche stellt (empfohlene Mindest-Grafik Ausstattung: RTX 4080), wird anhand einer endlos erscheinenden Urwaldlandschaft demonstriert, was PCG leisten kann. Wer den Waldrausch auf den Gipfel treiben will, kann dies auch mit einer VR-Version tun, die der umtriebige Youtuber JSFILMZ auf Artstation ([is.gd/artstation_UE_dreams_demo](https://www.artstation.com/ue_dreams_demo)) mit zusätzlichen Details wie Tieren ausgestattet hat.

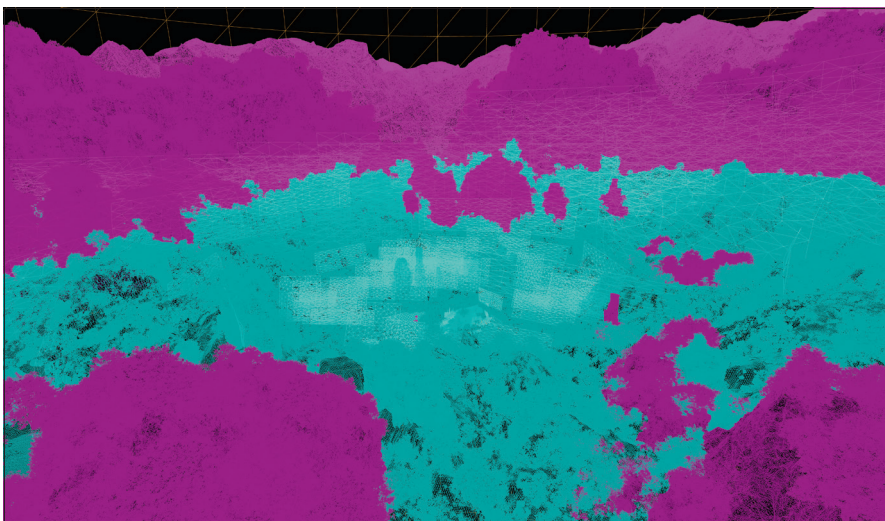
Selbstverständlich wird auch hier ein sportlich ausgestatteter Rechner vorausgesetzt, denn das wildwüchsige Gestrüpp mit dichtem Unterholz auf riesigem Terrain hat es in sich. Dennoch ist der Urwald Wahnsinn aus erstaunlich überschaubaren Asset-Bausteinen aufgebaut, denn es reichten relativ wenige, aber dafür dynamische Quixel Asset Pakete, um die grüne Hölle zu bevölkern. Die besagten Quixel Assets sind jeweils in Assemblies verpackt, die man, wie man es seit Version 5.0 kennt, per Rechtsklick editieren kann. Möchte man nun die integrierten PCG-Komponenten der Szene anpassen, kann man das ganz einfach mit der integrierten Spline Kurve via dem Details Panel machen. Das heißt zum Beispiel via Alt-Drag einen Punkt hinzuzufügen, damit sich dieser Landschaftsabschnitt entlang der Kurve verlängert, oder nach belieben, die Punkte der



Der Electric Dreams Urwald in der Übersicht mit den alternativen Viewport Ansichten: Lighting only, Wireframe und Nanite Instances.

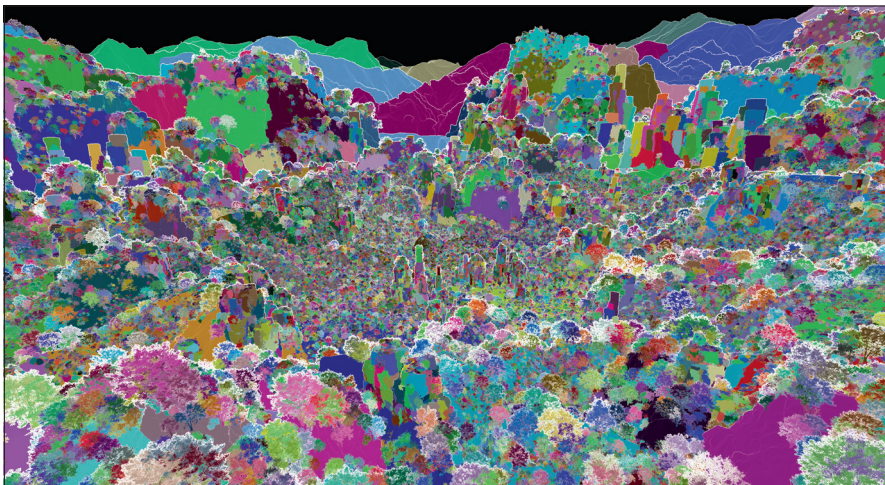


Kurve zu verschieben, kopieren oder zu duplizieren. Diese sogenannten „Packed Level Actors“ sind also, trotz komplexer Logik dahinter, intuitiv zu bedienen. Parameter wie Seed und Density sind leicht zugänglich und zusätzliche Mesh Features lassen sich ab- und anschalten. Wenn man einmal schnell einen anderen Look ausprobieren will, braucht man nur die Scatter Presets auszutauschen und schon ist die Umgebung optisch angepasst. Fazit des Electric Dreams Demos: Eine atemberaubende Wald Expedition mit Unreals mächtigen neuen PCG Tools, die einen erahnen lassen, wo die prozedurale Reise hingehet.

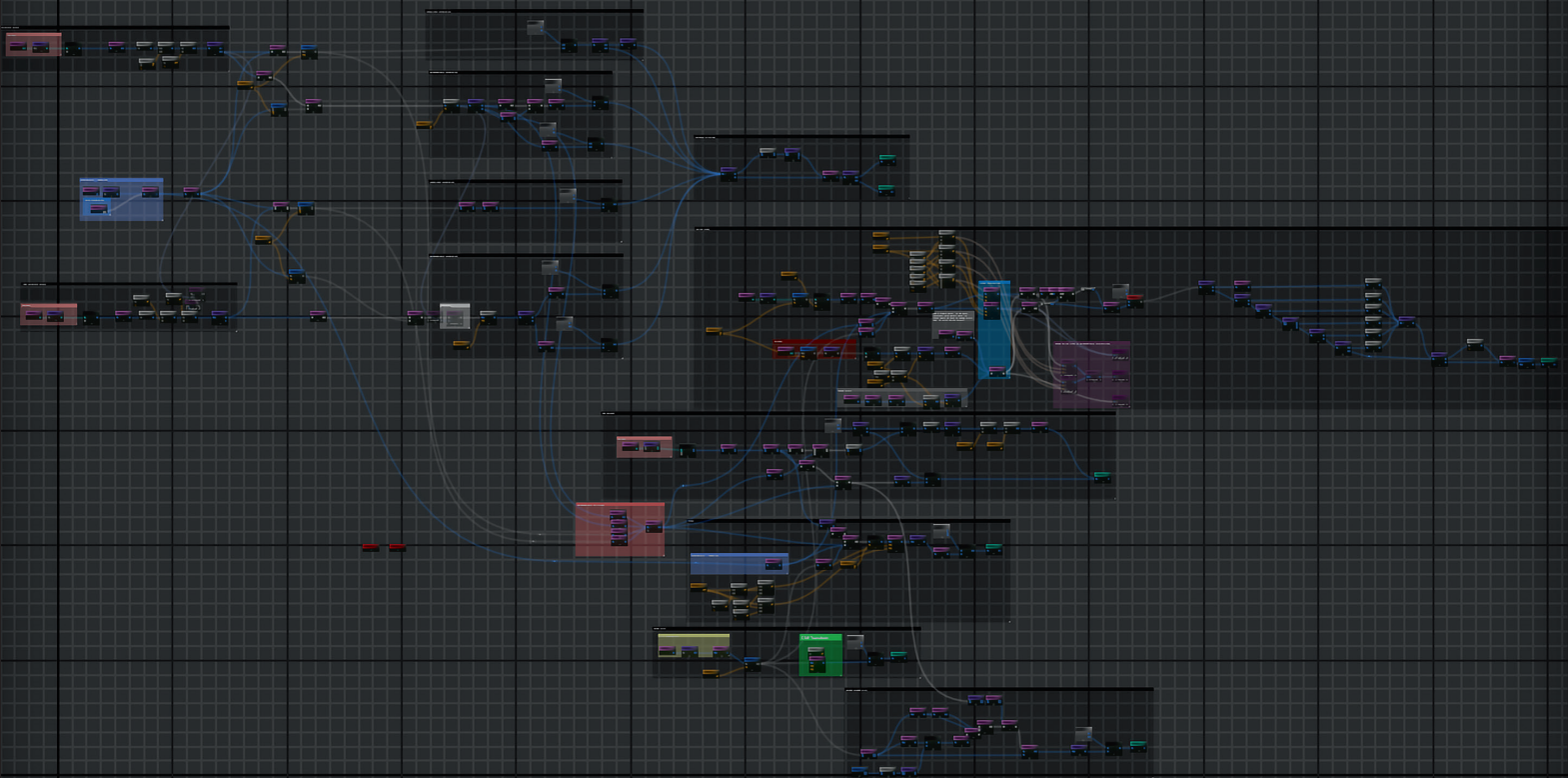


Der parametrische Landschaftsbau Kasten

Ein schnelles Aha-Erlebnis für die praktische Anwendung verschafft man sich am einfachsten, wenn man mit einer frischen Szene anfängt und ein typisches Use Case ausprobiert, z.B. eine Landschaft mit einem Gebäude umringt von dichtem Wald. Also, innerhalb des Electric Dreams Projektes via File eine neue Modular Map erstellen, alle Landschafts-Kacheln mit Landscape Material versehen und via Content Browser den Blueprint Actor PCGDemo_ForestBP in die Szene platzieren. Die ganze Magie entfaltet sich nun, wenn man nun unter dem Details Panel der PCG-Sektion den Button Generate aktiviert.



Erwartungsgemäß wird nun das Terrain mit den Wald Assets bestückt und es steht einem offen, die ein oder andere Entstehung nachzubessern. Um nun eine Aussparung für das Gebäude zu generieren, ist ein weiterer Blueprint Actor notwendig, den man ebenfalls über den Content Browser findet: Der PCGDemo_DitchBP Actor. Platziert im gewünschten Bereich hat dieser nun ein Spline Kurve im Detailfenster, mit der sich nun die Aussparungs-Grenze zum Waldwuchs genau bestimmen lässt. Nachdem das Gebäude platziert ist, zum Beispiel zu Demonstrationszwecken das Medieval Castle Asset von Game Asset Factory, das sich schnell modular zusammensetzen lässt, kann man nun die Land-



PCG	
Enable Cliff	<input checked="" type="checkbox"/>
Enable Landslide	<input checked="" type="checkbox"/>
Enable Fallen Trees	<input checked="" type="checkbox"/>
Plants Max Scale	2.5
Plants Min Scale	1.0
Trees Radius	150.0
Trees Seed	50
Embankments Seed 1	55405
Embankments Seed 2	45
Fallen Trees Seed	2
Fallen Trees Probability	0.4
Fallen Trees Near Reach...	-390.0
Fallen Trees Far Reach D...	-500.0
Fallen Trees Spacing	500.0

PCG Demo User-Parameter im Detailfenster und der PCG Graph eines Assemblies.

Aufforsten, bei Null angefangen

Um PCG Actors ganz neu aufzubauen, muss zunächst der PCG Plug-in aktiviert sein. Eine neue Szene mit PCG Graph, den man als Volumen in der Landschaft platziert, erstellt man dann zum Beispiel via dem PCG-Eintrag im Content-Browser Rechtsklick-Menü. Als zusätzliches Element kommt nun der Blueprint Actor dazu, der wiederum eine Spline Kurve als Komponente enthält, die jetzt via Tag angesprochen werden kann.

Nach Blueprint Manier werden nun im PCG Graph die entsprechenden Nodes verknüpft, um aus den räumlichen Szenen-Information Daten abzugreifen und zu verarbeiten. Zum Beispiel via „Get Landscape Data“ oder auch „World Ray Hit Query“ (um zum Beispiel mehrere lose Objekte aus einer Richtung zu treffen), oder wie in dem gerade erwähnten Beispiel mit „Get Spine Data“ Landschafts-Punkte aus den Kurven Informationen zu generieren, diese zu modifizieren um dann die Mesh Objekte an den neuen Koordinaten auszuspucken. Das lässt sich schnell erreichen in dem man über Get DSpline Data einen Blueprint Actor mit einem eigenen Tag (hier DPdemo) aufruft und via Spline Sampler, dessen Spline-Info

abgreift, um dann die daraus generierten Punkte via Transform Points zu verfeinern und dann an die Static Mesh Spawner Node auszuliefern, wo schließlich das Mesh Objekt erzeugt wird.

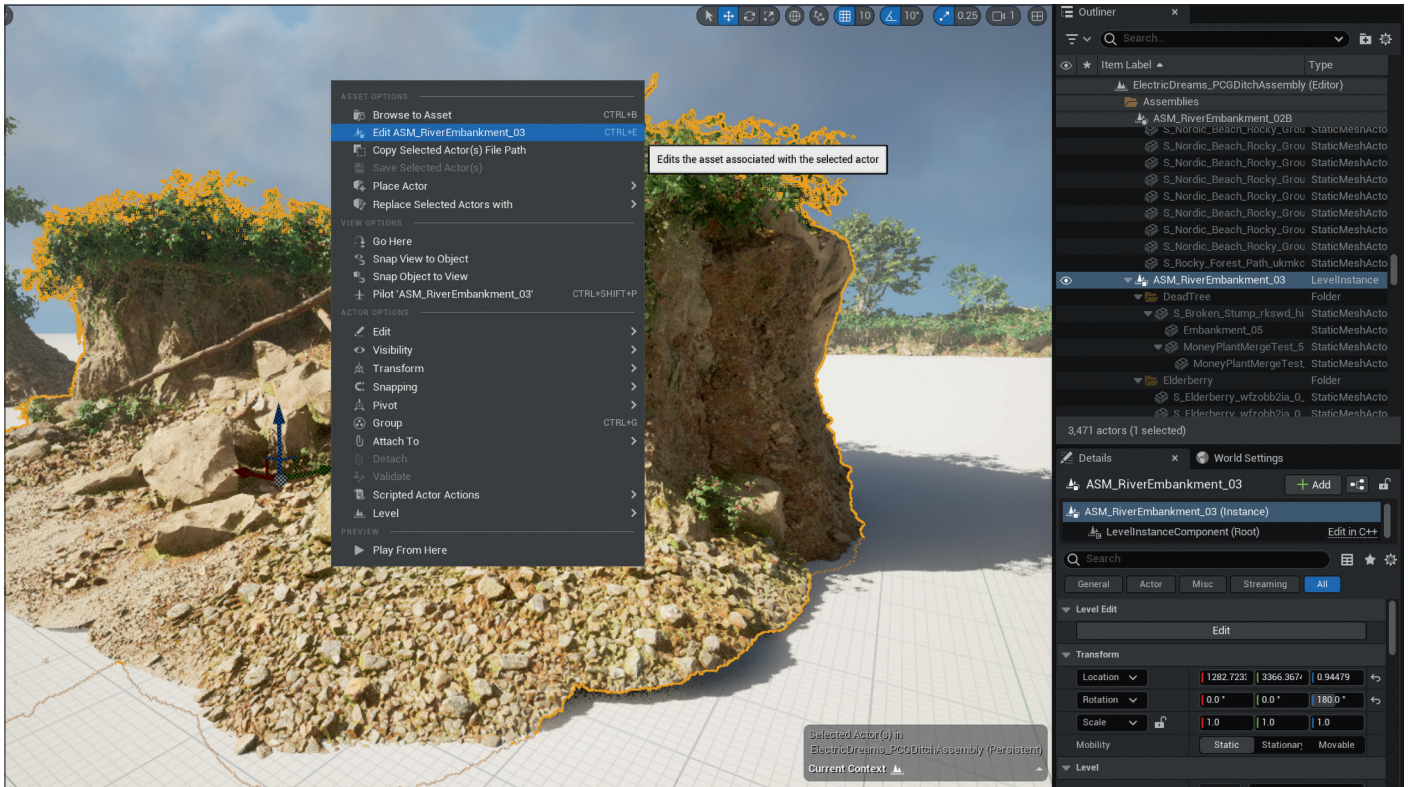
Glücklicherweise kann man über eine Debug Funktion mit Keyboard Shortcut d, im PCG Graph jeweils die Debug Visualisierung per Node aufrufen und via Shortcut e, die einzelnen Node temporär deaktivieren. Das hilft enorm, um dann mit der groben Debug Punkte Ansicht zu analysieren, wo die Punkte genau hinfallen, um dann schnell nachzubessern, ohne dass jedes Mal die ganze Landschaft generiert werden muss.

Um die verschiedenen Landschaftselemente wie z.B. Bäume, Büsche und Grasbüschel, etc. zu generieren und zu verteilen, gibt es verschiedene Wege. Zum Beispiel kann man über Filter-Funktionen, Punkte zur Generierung ausfiltern, PCG Graphs via Subgraphs verschachtelt organisieren, oder einfach neue Ableger in dem PCG Graph anlegen, die dann jeweils ihre eigenen Funktionen haben und die zusätzliche Mesh-Typen generieren.

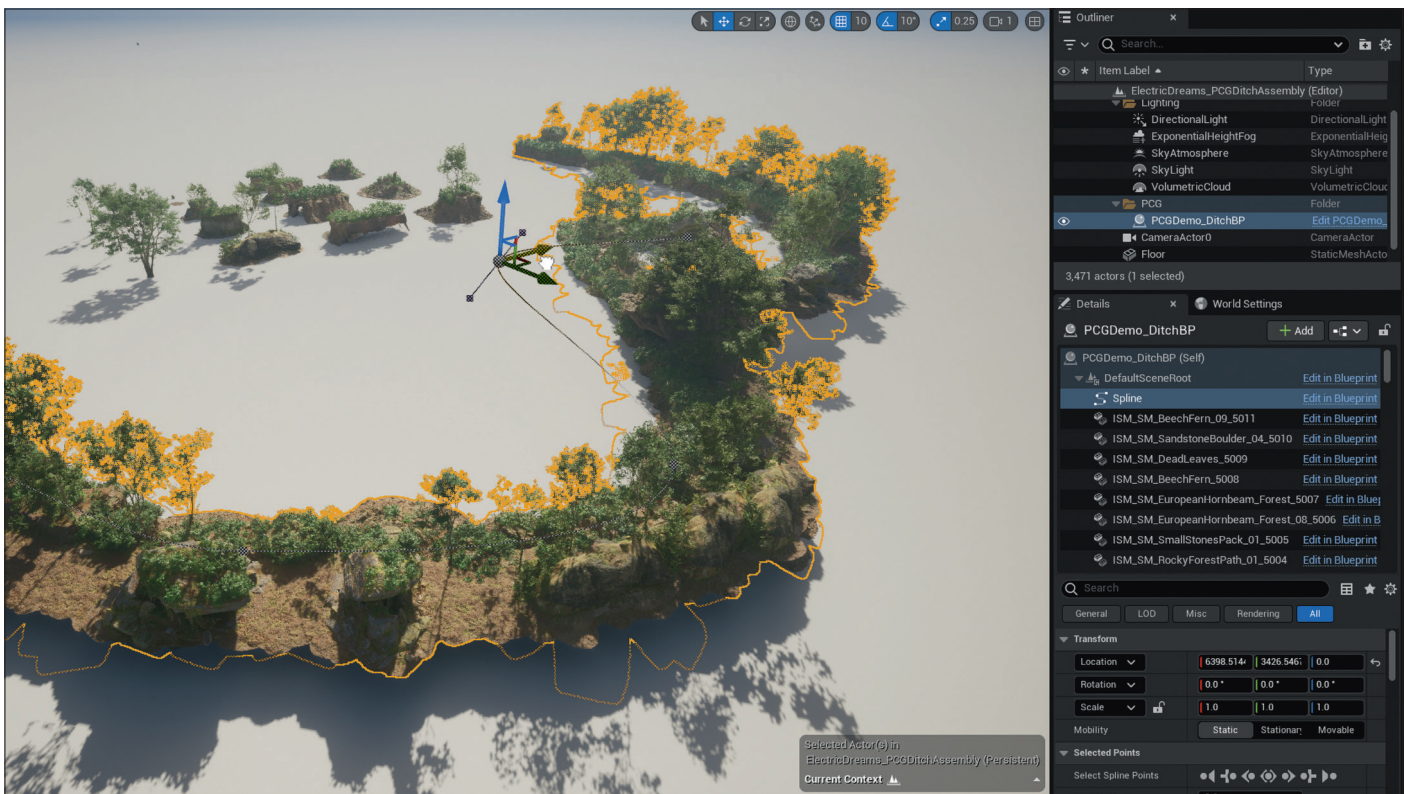
Wenn man nun zusätzliche Landschaftsfunktionen einbauen möchte, kann dies über weitere Blueprint Actors mit Spline Compo-

nschaft präzise anpassen. Das Schöne an diesem Vorgehen ist, dass jetzt via Spline Kurve genau bestimmen lässt, wie eng der Wildwuchs an den Mauern liegen soll, bzw. wie viel Platz das Gemäuer in der Komposition braucht.

Mit dem Zusammenstellen der PCG-Bauteile zeigt sich der Produktivitätsgewinn, der hier erreicht werden kann. Hätte man dieses Gestrüpp wie vorher per Hand eingefügt oder via Asset-Painter gemalt, wäre der Zeitaufwand ein Vielfaches gewesen.



Die PCG Bausteine: Quixel Assets als Compound und dann als „Packed Level Actors“ Blueprints, die sich via Details Parameter und Spline Kurven steuern lassen.

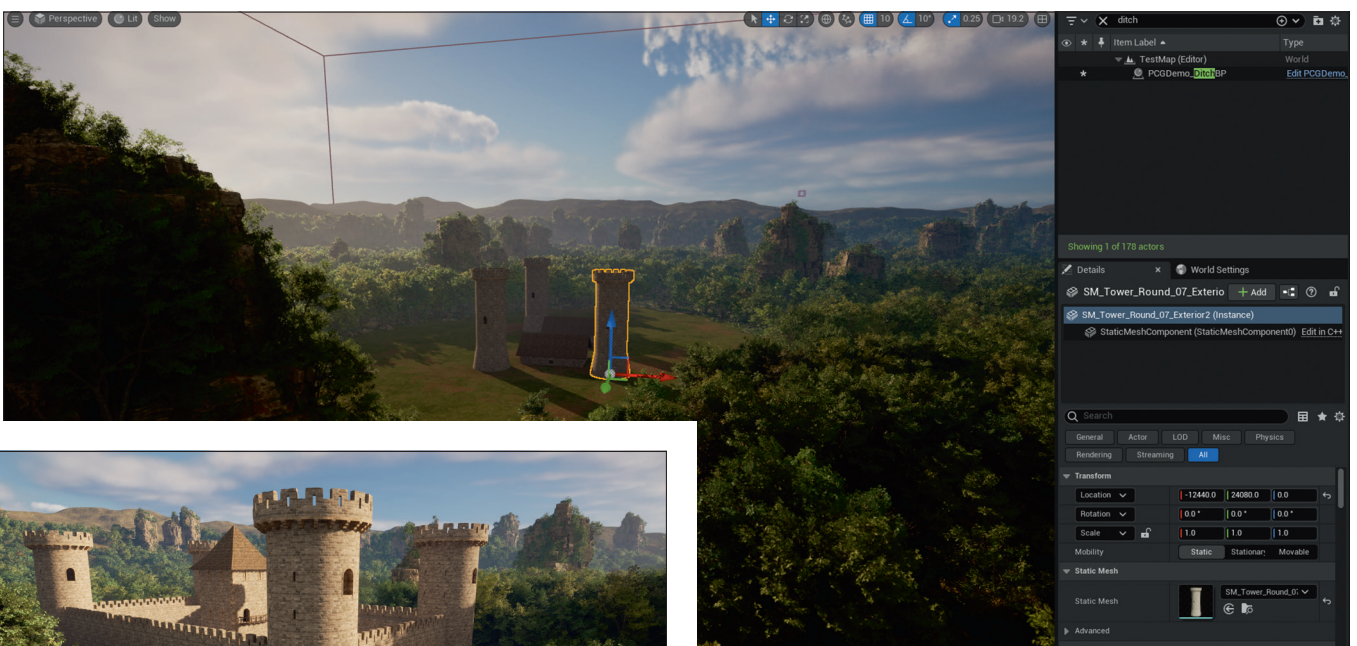
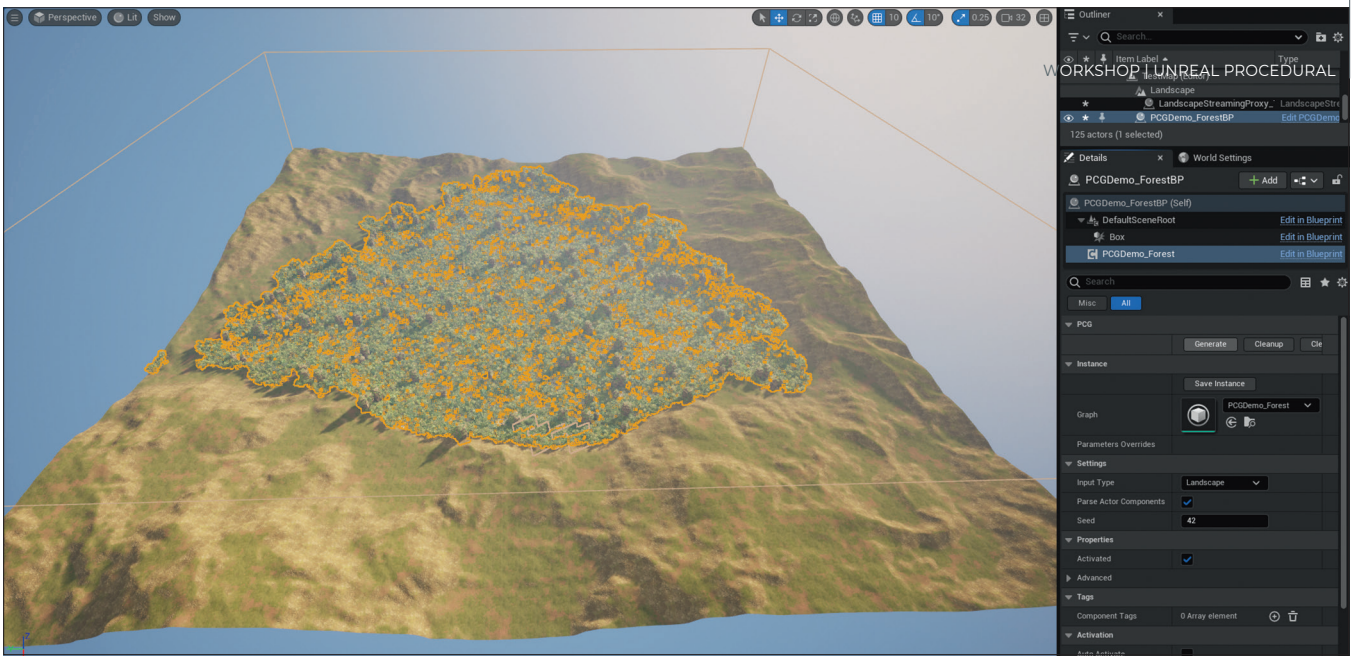


ment erreichen. Zum Beispiel einen Pfad durch den Wald freimachen, in dem dieser neue Spine Actor im PCG Graph via Tag aufgerufen wird und via der Bounds Modifier Node in der Breite ausgedehnt wird, damit die daraus resultierende Punktmasse dann von der vorherigen via Difference Node ab-

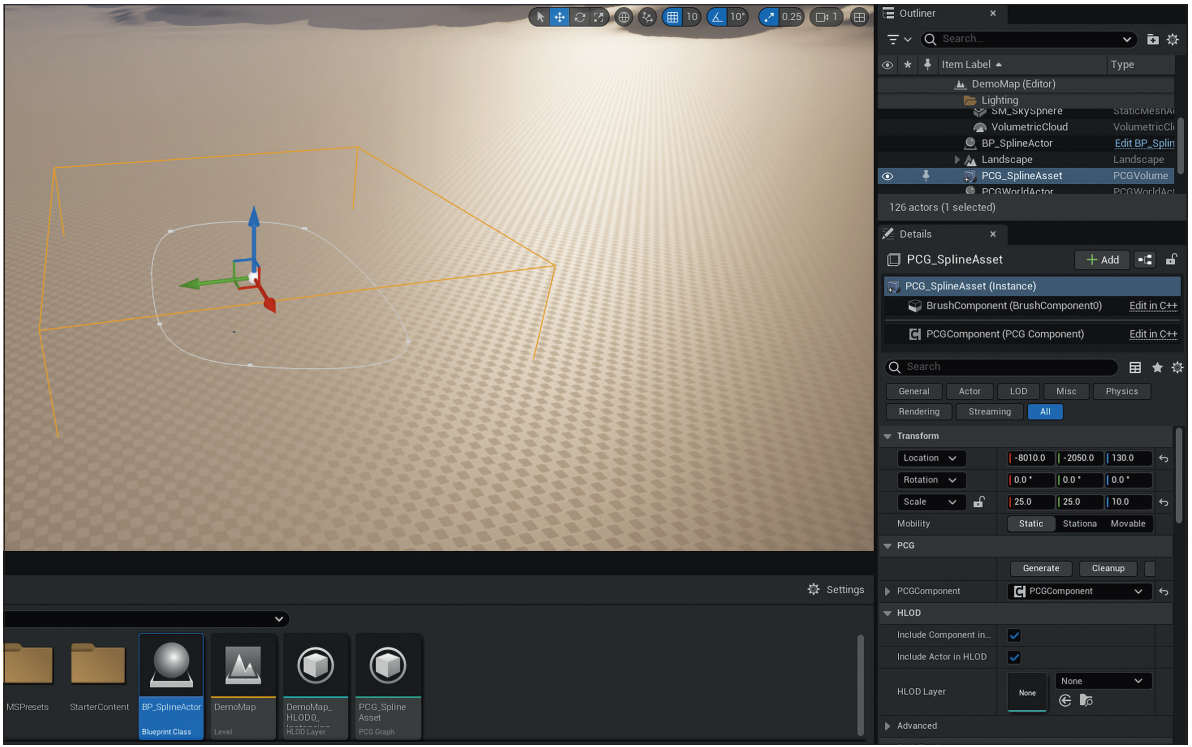
gezogen werden kann. Das Resultat ist dann der besagte Weg durch das Gestrüpp, der sich, ebenso wie das Waldstück, dynamisch modifizieren und ausweiten lässt.

Andere Komponenten aus dem Unreal-Landschaftssystem lassen sich natürlich bestens mit PCG Graphs kombinieren, wie zum

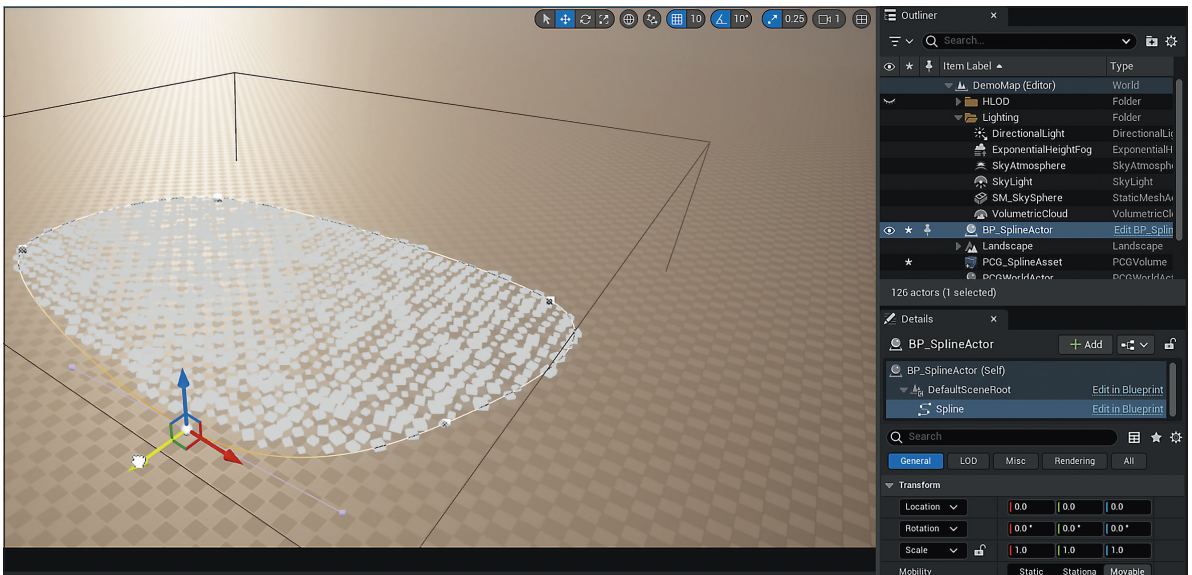
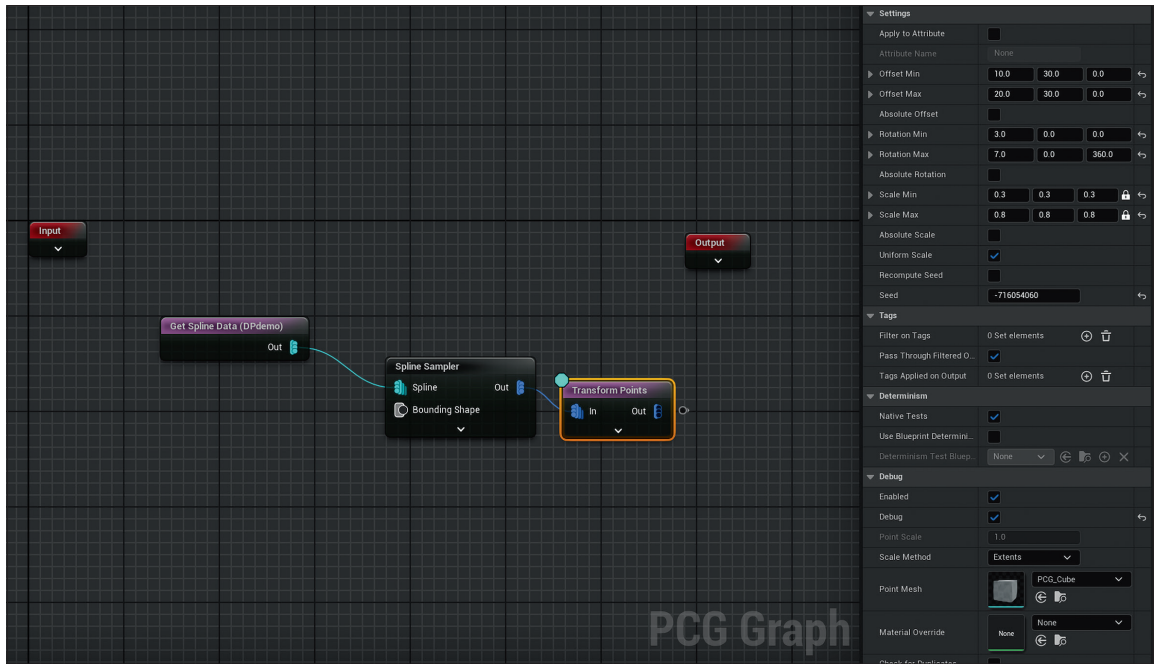
Beispiel die Fluss, See und Ozean Elemente des Unreal Water Plug-ins. Einmal aktiviert, stehen nun die Wasser Landschaftsteile zur Verfügung und da See und Fluss über eingebaute Splines verfügen, um die Form zu gestalten, sind diese natürlich auch über den PCG Graph abgreifbar.

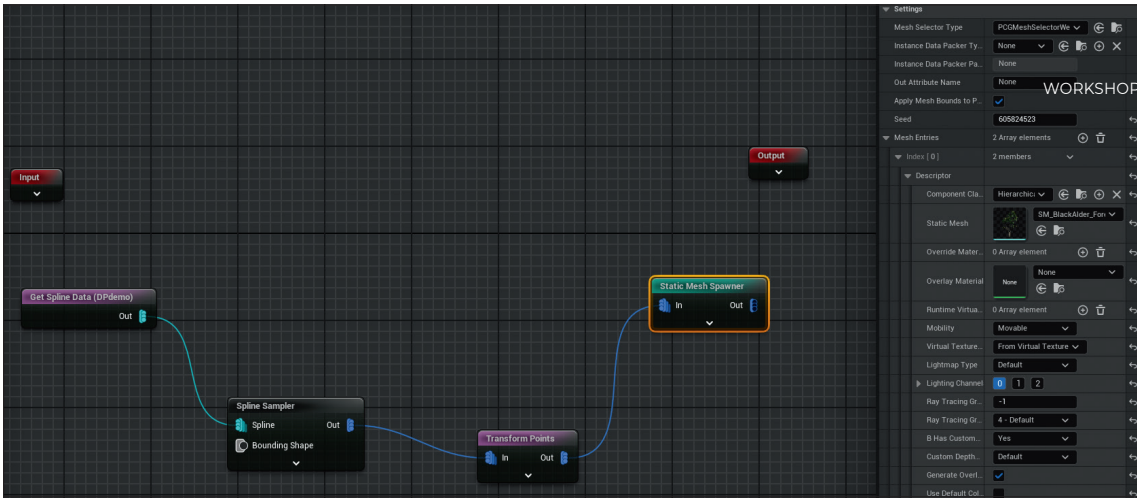


Ein PCG Szenenaufbau im Schnelltest mit Assets aus dem Electric Dreams Demo. Wald anlegen, Aussparung via Spline definieren, modulare Gebäude hinzufügen. Fertig ist die Burg mit Wald Umgebung.

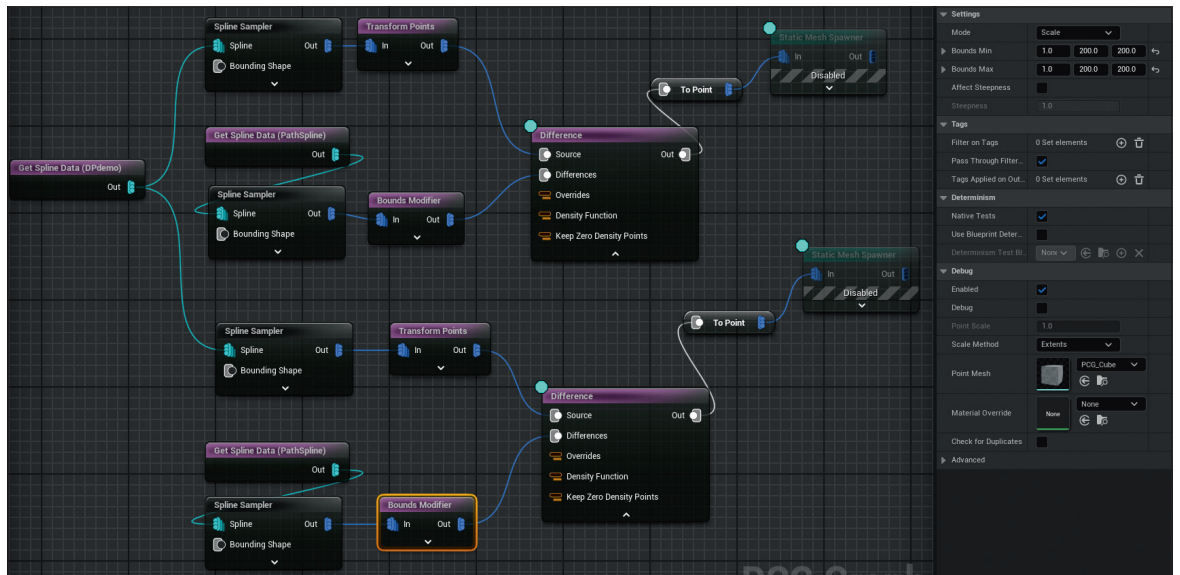
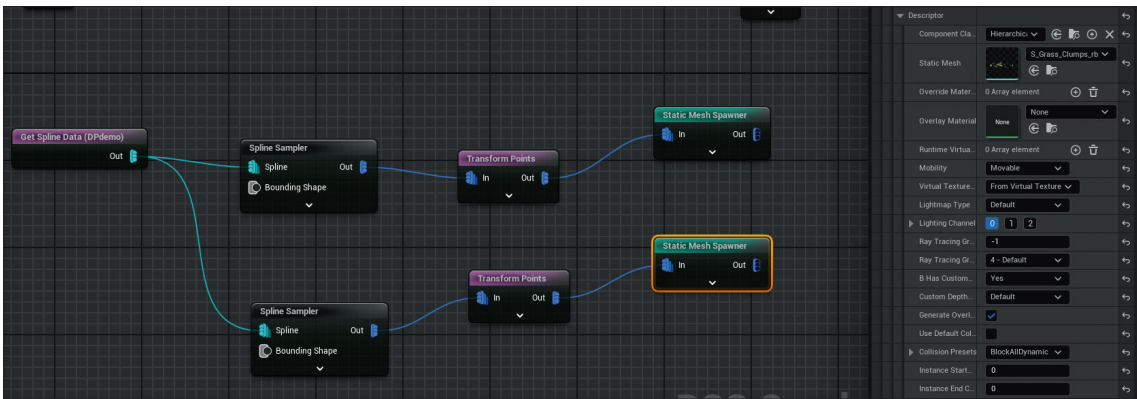


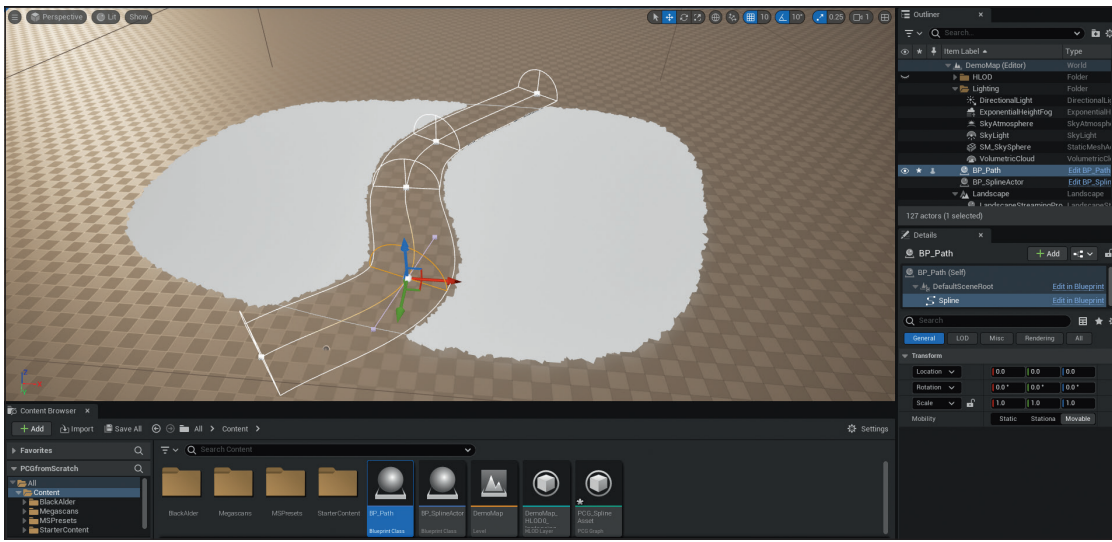
PCG von Anfang an: PCG Volumen und Blueprint Actor mit Spline, Vorsicht der Punkteverteilung via Debug Mode, und dann – wie ihr auf der nächsten Seite seht...



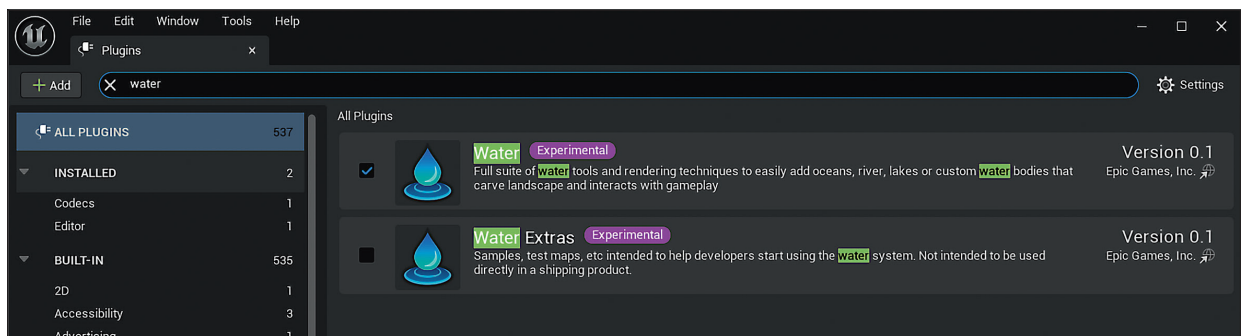
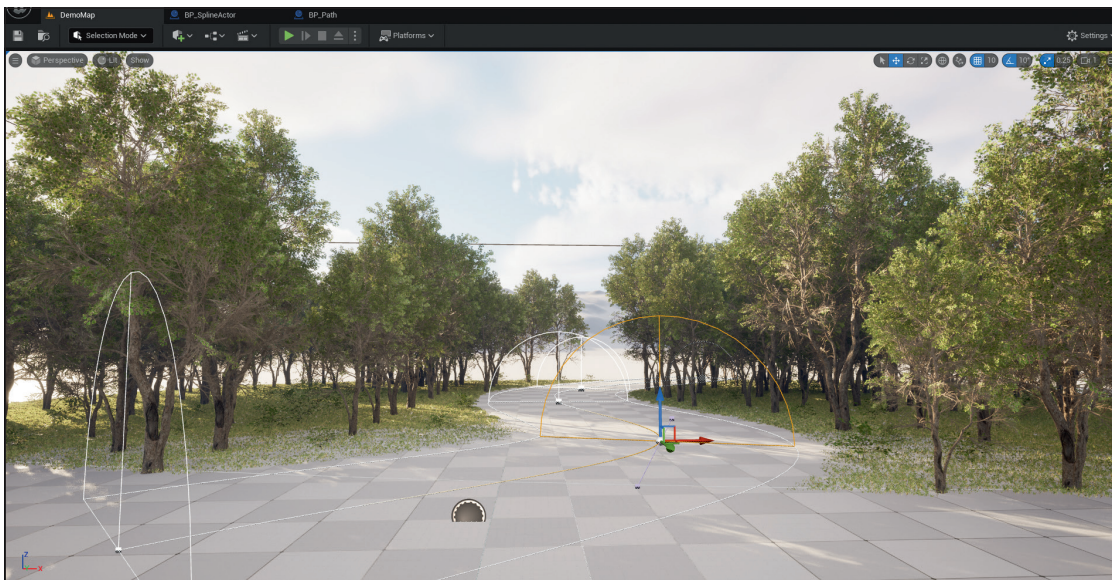


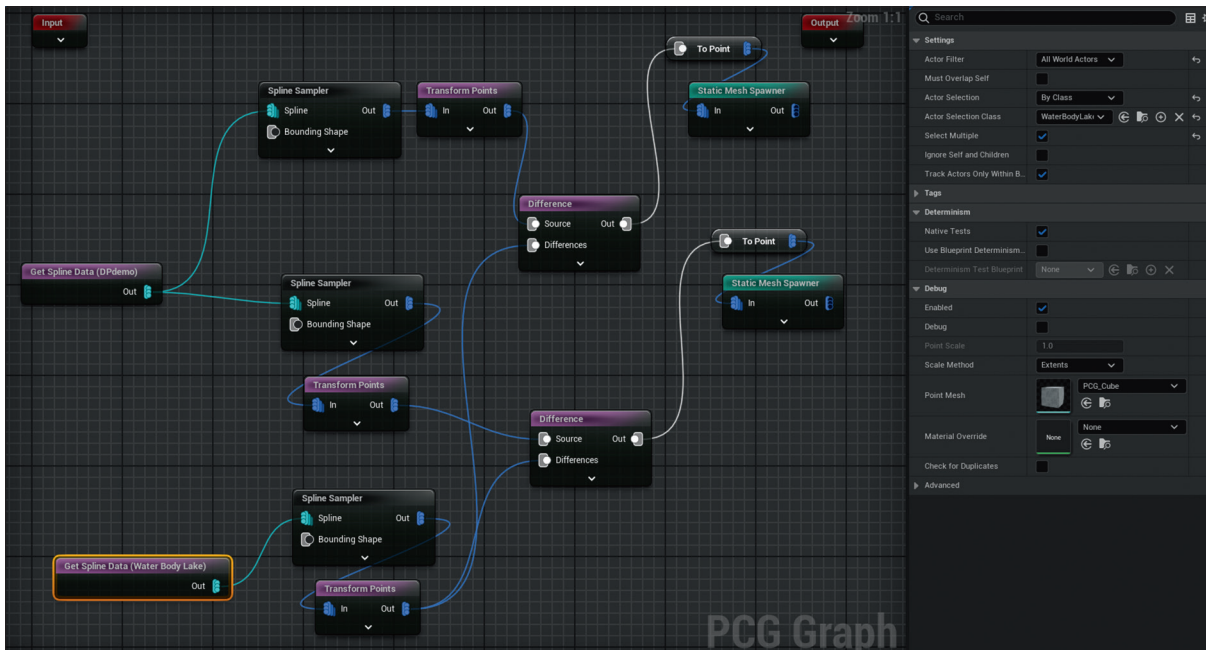
Quixel Alden-Bäume als Mesh Ausgabe an den Punkte-Positionen. Dann hinzugefügt eine Verzweigung für Gras-Detail und kleines Gebüsch.



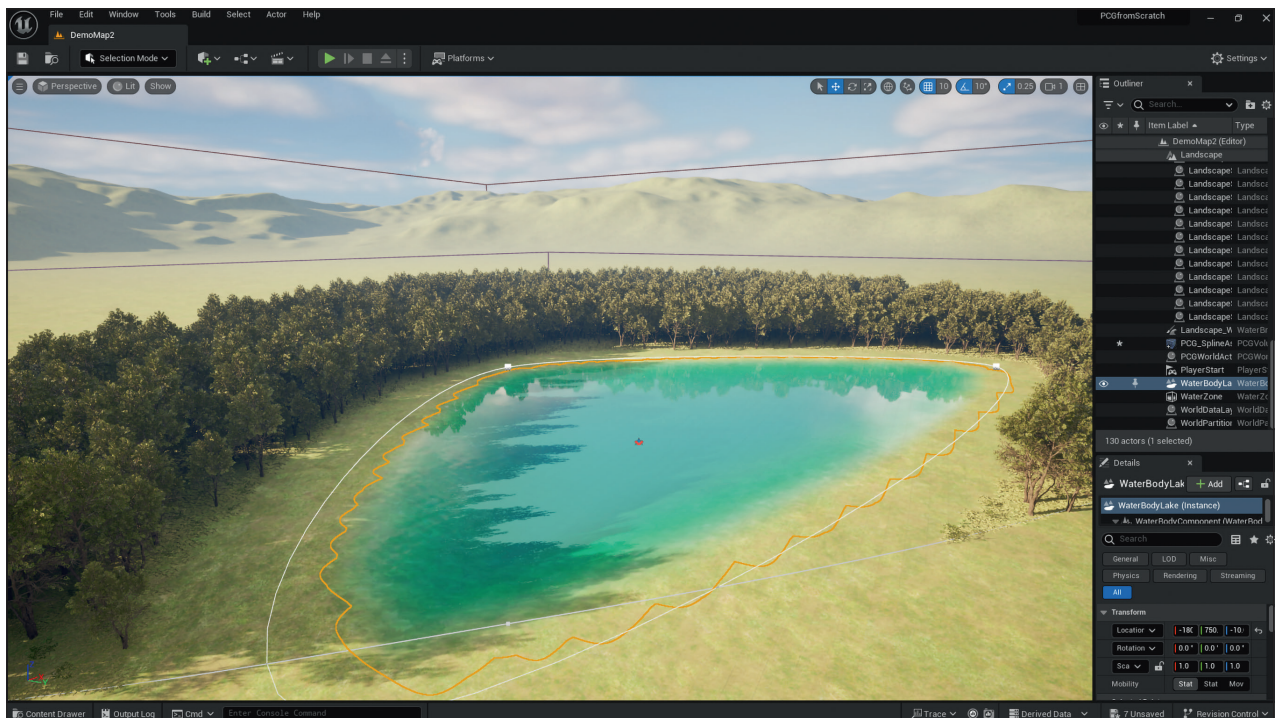


Um dann einen Weg dazwischen freizumachen, wird ein Blueprint Actor mit Spline und Bounds Modifikation via Difference Node von der vorherigen Punkte Menge abgezogen.





Um einen See, via Unreals Water Plug-in, mit der dazugehörigen Spline anzusprechen, genügt es, den Wassertümpel via Class Selection auszuwählen und ebenso via Difference den Wald zurückzuschieben.



Wenn man zum Beispiel nun einen See (Water Body Lake) in die Szene platziert, kann man diesen nun im PCG Graph via get Spline Data aufrufen, in dem man als Selektionsmethode All Actors of Class und im Spline Sample On Interior definiert. Die generierte Punkte-Masse muss nun via Transform Points-Skalierung etwas ausgedehnt und wie vorher subtrahiert werden, sodass das Waldstück in der Landschaft dann dynamisch zurückgedrängt wird.

Mit dieser Methode können nun recht komplexe Konstruktionen gebaut werden, wo verschiedene Spline und Parameter dy-

namisch steuerbar aufeinander einwirken, um neben Density, Scale und Feature Aktivierung auch Komponenten-Gewichtung zu steuern. Das eingangs besprochene Electric Dreams Demo ist dafür ein gutes Schaulaufen, denn die PCG Graphs von Epic Games demonstrieren, dass in Sachen Komplexität nach oben alles offen ist.

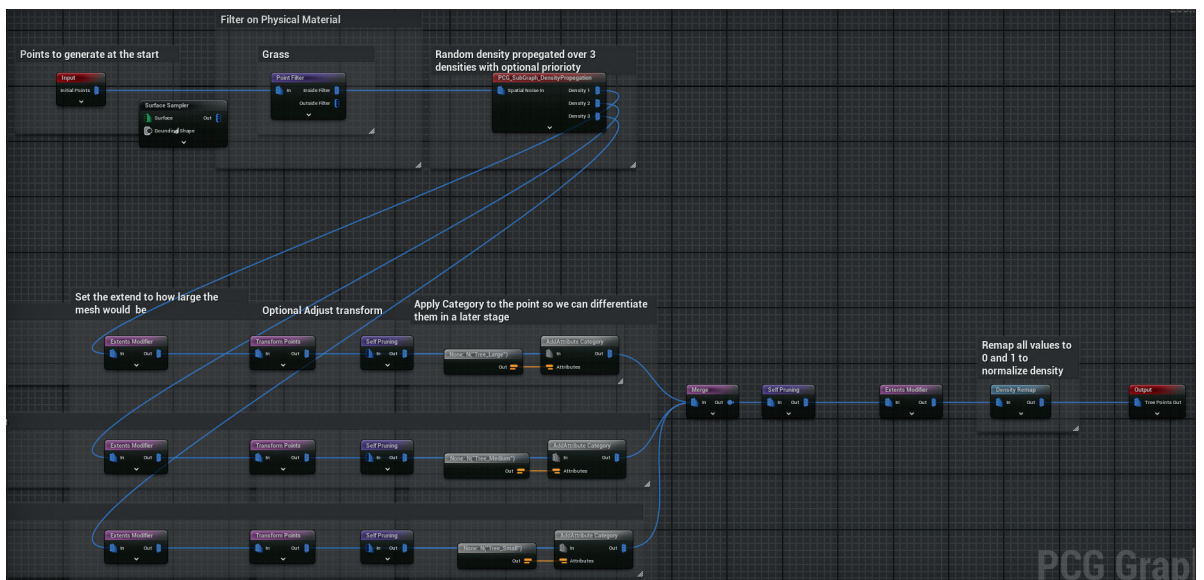
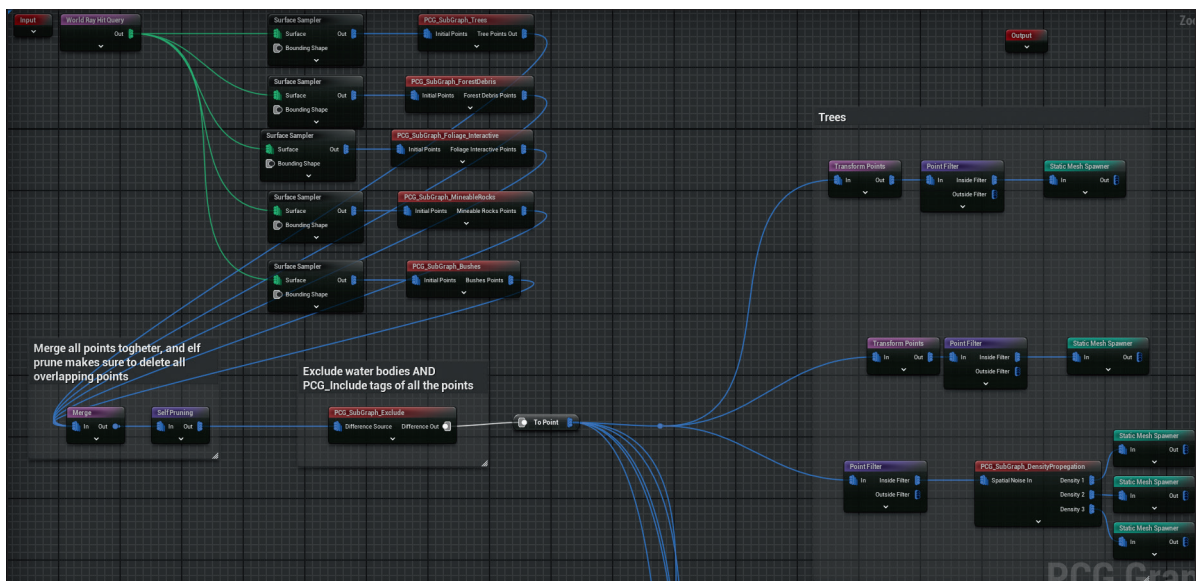
Punkte Menge abgezogen. Um einen See, via Unreals Water Plug-in, mit der dazugehörigen Spline anzusprechen, genügt es, den Wassertümpel via Class Selection auszuwählen und ebenso via Difference den Wald zurückzuschieben.

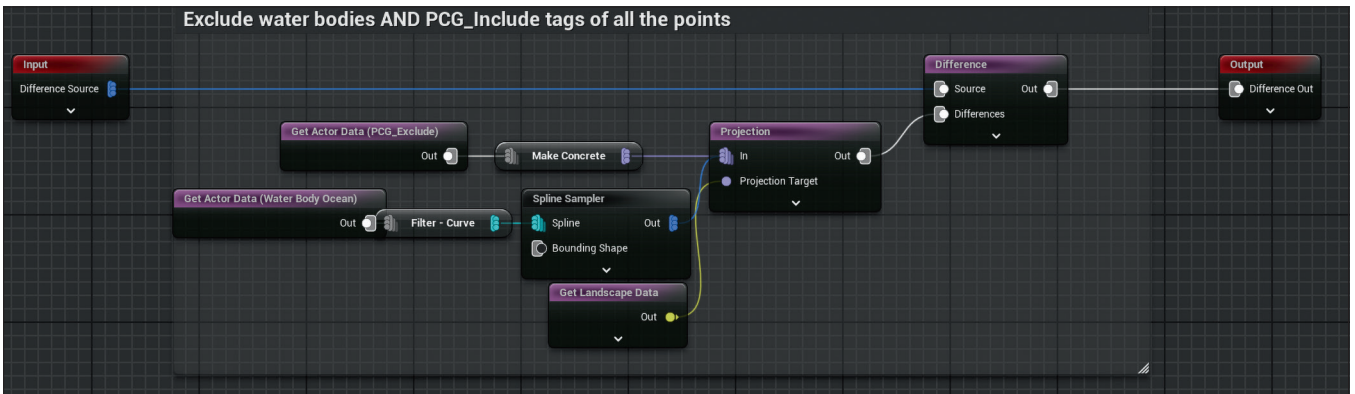
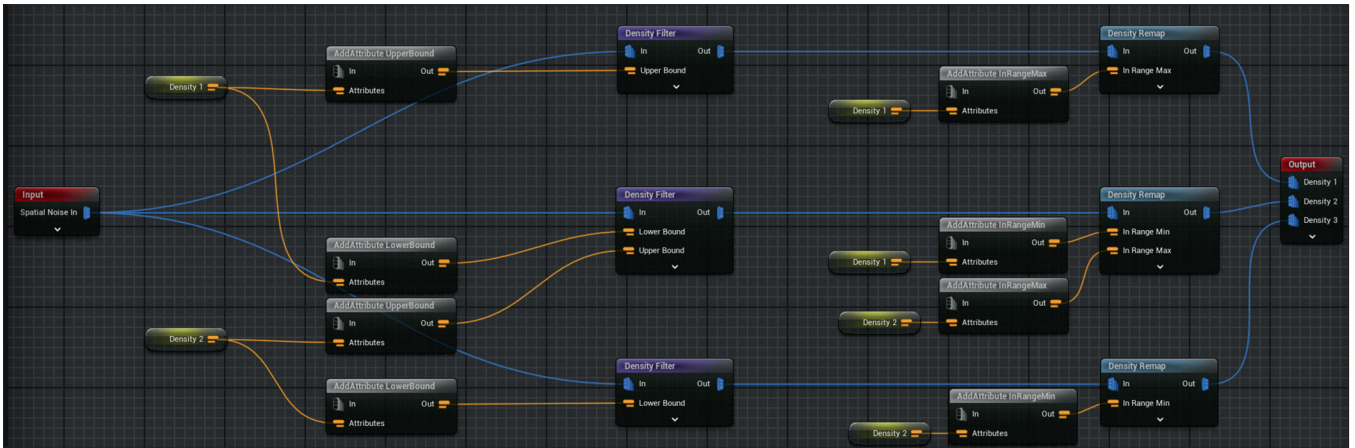
Geht auch: PCG Konstruktionen für Bauelemente

Selbstverständlich funktionieren PCG Graphs auch für die Generierung von menschengemachten Strukturen wie Mauern, Straßen und Gebäuden. Da das System noch recht neu ist und die Dokumentation bisher eher spärlich ist, ist man hier auf Beispiele von Unreal Users angewiesen, die sich in die Bresche geschlagen haben und durch Herumprobieren Lösungen entdecken konnten. Es ist, wie gesagt, zu erwarten, dass dieser Bereich in naher Zukunft besser



Der „Basic Forest“ von Hyper in der Insel Demo. Mit einem „World Ray Hit Quarry“ werden die Surface Informationen wie Felsen- oder Grasfläche via Unreal Physical Material abgerufen. In dem PCG Haupt-Graph durchläuft der Datenfluss die verpackten Subgraphs, inklusive des Exclude Subgraphs, bis die Punkte durch Filter-Transformation zur Mesh Generierung durchgereicht werden. Durch die häufig verwendeten Node „Self-Pruning“ werden dabei Überlappungen ausgeschlossen. In den nachfolgend gezeigten Subgraphs sieht man wie die Daten aus der „Density propagation“ geformt werden, die Custom User Parameter „Density“ zur Steuerung eingebaut und die exkludierten Landschaftsteile, via Difference aussortiert werden.



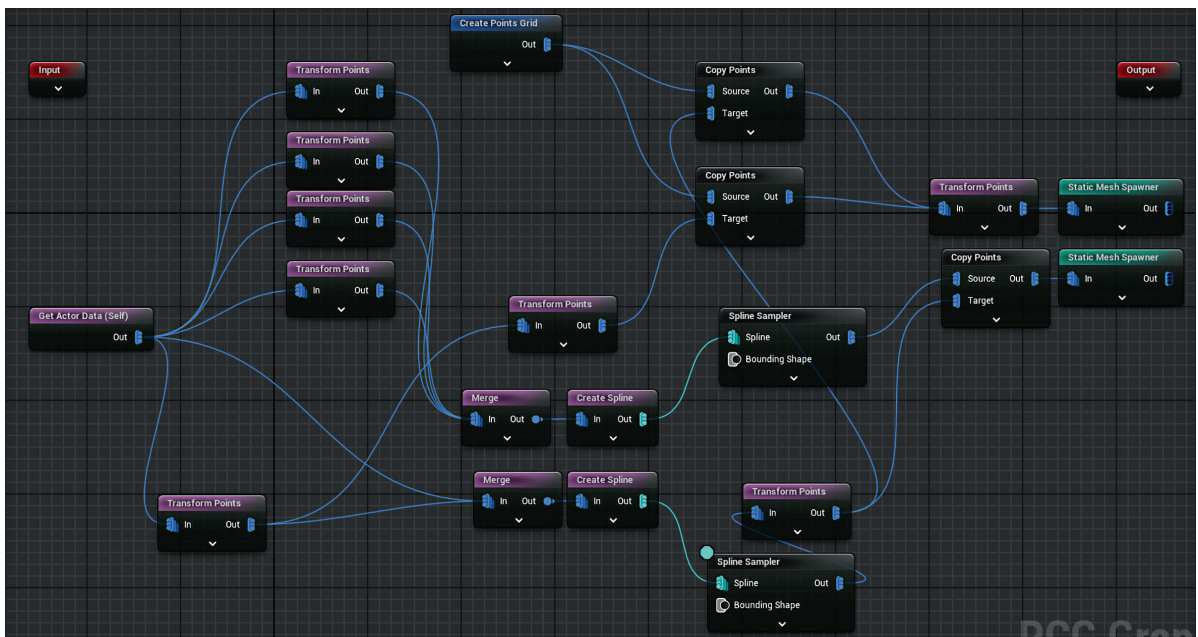
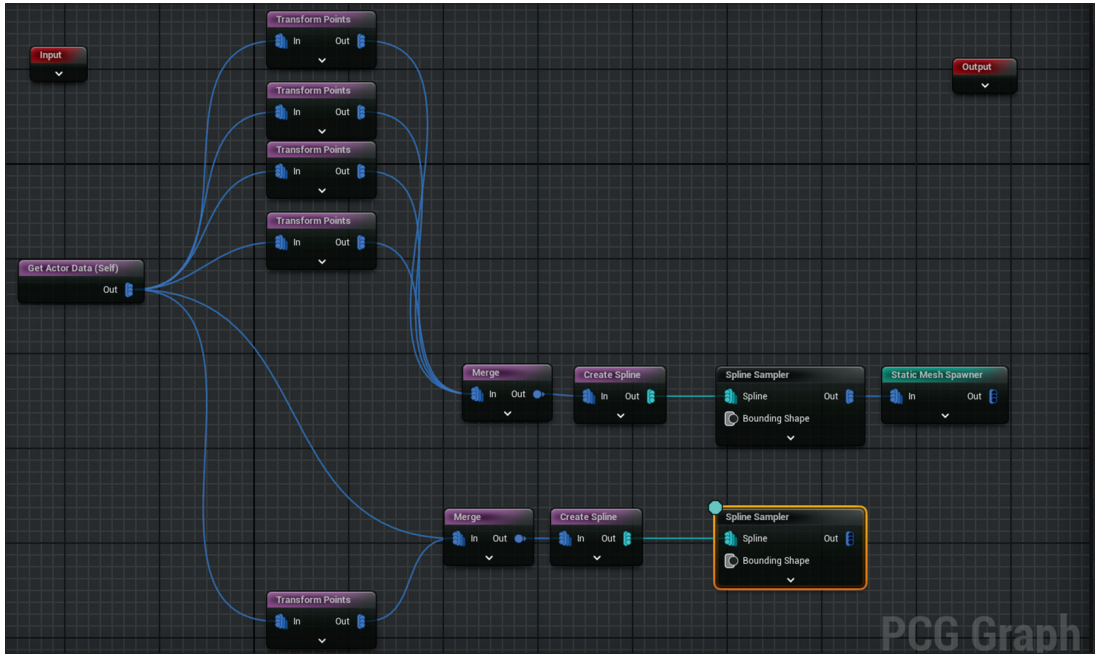


Die Subgraphs Density Propagation und Exclude: Hier sieht man wie die Daten aus den Bounds, Filter und InRange Nodes ge-remapped und die exkludierten Landschaftsteile, via Difference, aussortiert werden.



Hyper's Basic Forest PCG in einem Custom Environment mit Quixel Assets ausgeschmückt. Das Gebäude ist diesmal via Blender, mit dem Procedural Building Generator 2, entstanden.

In diesem Beispiel generiert der PCG Graph aus der World Position des Blueprint Actors ein Punkt, der dann via Transform Points multipliziert und positioniert wird, um die Grundfläche zu bilden. Das Resultat wird gemerged, um eine Spline zu erstellen, aus der dann via Spline Sampler die Meshes generiert werden. Aus den Meshes des Gebäudes-Segments ergibt sich dadurch das Erdgeschoss.





Mit einem Punkte Offset an der Z-Achse entstehen nun die Geschosse und das oberste Dach mit Struktur.

dokumentiert und auch einfacher zu bedienen sein wird.

Vom Prinzip her ist es bei Gebäudestrukturen nicht anders als bei Landschafts-Vegetation: Es werden Punkte generiert, die nach Parametern gefiltert, multipliziert und gesteuert werden, und dann am Ende als Mesh platziert werden. Zum Beispiel als PCG Component in einem Blueprint Actor, der in der Map positioniert wird. In dem PCG Graph wird dann via Get Actor Data, der erste Punkt in der World Position generiert, dieser dann mehrfach via Transform Points kopiert, verschoben, gemerged, via Spline Sampler neu eingeteilt und dann als Static Mesh Component an den Endpositionen generiert, so dass aus den Starter Content Wand-Stücken (Wall with Window) nahtlose Gebäudestrukturen werden, die sich dann via Offset Werten, als neue Stockwerke aufsetzen lassen.

Dieses Prinzip lässt sich selbstverständlich fortsetzen, indem weitere Gebäude-Cluster und ganze Städte mit zusätzlicher Offset-Werten und Variationsparametern generiert werden. Da wird sicher noch einiges kommen.

Marketplace PCG's: Designer Assets mit Benefits

Nicht immer hat man die Zeit, einen flexiblen und gut strukturierten PCG Graph selber

aufzubauen. Ein Shortcut ist, sich Konzepte oder Inspiration aus dem PCG-Bereich des Blueprint Web Depository blueprintue.com/search zu holen, bzw einfach per copy/paste das eigene Projekt einzufügen. Ein anderer Weg ist, auf Shopping Tour bei kommerziellen PCG Anbietern zu gehen. Vorbildlich strukturierte und dokumentierte PCG Assets gibt es zum Beispiel von dem Unreal Marketplace Anbieter Hyper. Der Basic Forest von Hyper (is.gd/marketplace_basic_forrest) ist elegant aufgebaut, indem die entscheidenden PCG-Elemente in überschaubare PCG Subgraphs verpackt sind. Dadurch sind Aufbau und Funktion schnell erfassbar und einfach zu editieren. So kann man schnell die Density der einzelnen Landscape Elemente nachbessern und es ist sofort ersichtlich, wie das Resultat zustande kommt.

Das Asset Pack wird mit den entsprechenden Elementen, Bäume, Büsche, Felsgestein, bis zu Kleinteilen wie Pilzen und Debris geliefert. Nach den parametrischen Vorgaben des PCG Graphs werden diese über die Landschaft verteilt, wobei Wasser sowie Felsuntergrund ausgespart werden. Die Demommap mit einer bewachsenen Insel präsentiert die volle Pracht der PCG Elemente und bei einem kleinen Rundgang kommt echte Freude auf. Die Preset Assets lassen sich natürlich beliebig mit eigenen

Assets swappen, um sie eigenen Szenen anzupassen.

Fazit

Mit dieser Art von benutzerfreundlichem PCG Assets zeigt sich auch, wo die Zukunft liegt. Das betrifft insbesondere Unreal User, die schnelle Resultate brauchen und nicht unbedingt immer ein PCG-System von Grund auf neu aufbauen wollen. Zur Zeit sind jedoch noch relativ wenige PCG Assets im Unreal Marketplace vorhanden, das liegt sicher daran, dass das System noch zu neu ist. Hier wird sich sicher noch viel ändern, man kann damit rechnen, dass über die Zeit verschiedene Landschaften-, Gebäude- und City-Builder Assets-Lösungen auf PCG-Basis auftauchen werden. Die Zukunft ist Prozedural! > ei



Cornel Hillmann ist CG-Artist und XR Designer und seit über 20 Jahren im Bereich Media & Entertainment, Visualisierung und Design tätig. Er arbeitete unter anderem mit Marken wie Panasonic, Jaguar und Razor, führte 3D-Design-Kurse an der Limkokwing University und leitet Masterclasses für Immersive Media Postproduction, Advanced 3D-, VR- und Media-Design. Außerdem ist er Autor der Bücher „Unreal for Mobile and Standalone VR“ und „UX for XR“ des New Yorker Apress Verlags.