

Procesverslag low-poly tree generator

Casper van Battum | 3031249 | Procedural Art herkansing (2020-04-06)

<https://github.com/Creator13/ProceduralTrees>

Dit verslag beschrijft het proces en het ontwikkelen van een generator-tool voor procedurele bomen, in een low-poly stijl.

Vormkeuze

Vanwege mijn aantrekking tot alles wat de natuur te bieden heeft kwam ik op het idee om bomen te genereren. Deze bomen zouden in game-omgevingen en levels kunnen worden gebruikt om snel veel verschillende maar toch gelijksoortige bomen te maken. De bomen zijn specifiek in een low-poly stijl. Naast dat dit het uitwerken wat makkelijker maakt, is dit een stijl die mij in het algemeen aanspreekt. Het is simplistisch maar kan met een paar goede keuzes toch een goed uiterlijk opleveren. Hiernaast een paar afbeeldingen van het soort boom waar ik met deze generator naartoe gewerkt heb.



Realisme in L-systemen

Het eerste wat ik tegenkwam op het internet over het genereren van bomen was het fenomeen van L-systemen (Lindenmayer-systemen). Dit is een model ontwikkeld voor het modelleren van planten.

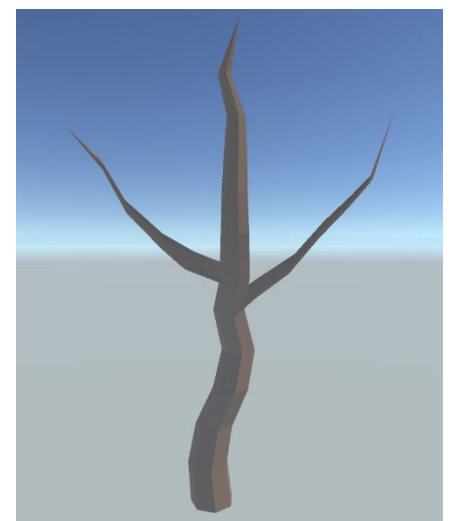
Daarnaast kunnen er veel patronen mee gemaakt worden. Ondanks dat het ontwikkeld is voor de biologie heeft de digitale kunst-wereld het ook opgepakt. Een groot deel van door de computer gegenereerde kunst werkt met L-systemen achter de schermen. Omdat het specifiek ontworpen is voor het modelleren van planten is het uiterst geschikt om bomen mee te maken.

Helaas zijn L-systemen vrij uitgebreid. Het zou ongepast zijn om een uitgebreide vertakking-structuur te maken voor een boom die uiterlijk twee tot drie zijtakken zal krijgen. Vandaar dat ik heb besloten dit model achterwege te laten voor een later project.

Simpele alternatieven

Er is geen direct alternatief model voor een L-systeem. Vandaar dat ik in het uiteindelijke project simpelweg trucjes heb toegepast die realisme suggereren.

De bomen 'groeien' volgens een aantal beperkingen. De stam is een apart onderdeel van de takken. De bladeren zijn weer een apart onderdeel. Door ieder segment van de stam of de takken telkens een klein beetje willekeurig heen en weer te bewegen krijg je een iets wat er redelijk uitziet als een echte boom, hetzij een wat kromme. Voor een oude eikenboom in een mysterieuze gestemde omgeving zou de boom in het voorbeeld hiernaast goed passen. Voor de bladerdakken kan een bol worden gebruikt die volgens perlin-noise wordt getransformeerd om willekeuriger te laten lijken.



Technische ontwikkeling

Het project is ontwikkeld in Unity 2019.3. De voornaamste reden hiervoor is dat ik me niet druk wil maken over het renderen van het model maar enkel over de vorm ervan. Een technische uitdaging in dit vak voor mij was het generen van een mesh vanaf nul. Hier had ik eerder al wat eenvoudige tutorials over gevolgd, maar tot nu had ik er nog nooit zelf mee geëxperimenteerd.

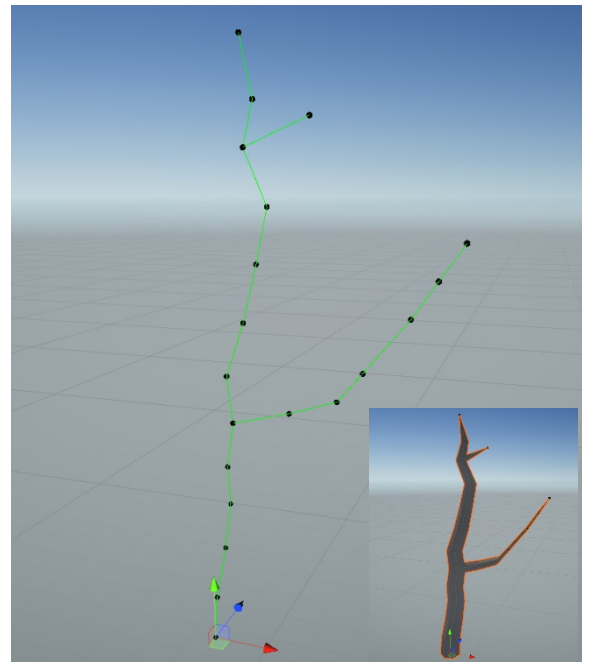
Opbouw en generator

Omdat er geen inherente samenhang in mijn boom zit zoals dat bij een L-systeem wel het geval zou zijn, heb ik gekozen om de elementen van de boom allemaal apart te houden. De verschillende elementen zijn de stam, de takken en de bladeren. Van de boom wordt eerst een skelet gemaakt. Alle punten van de stam en de takken worden vooraf berekend en daarna wordt het jasje van de mesh er omheen gelegd. Vervolgens zouden de bladeren gegenereerd moeten worden.

De gebruiker kan kiezen hoe hoog zijn boom wordt, hoeveel segmenten er in de hoogte zijn, uit hoeveel segmenten de cirkel van de stam bestaat en hoe dik hij is. Ook is er een waarde om aan te passen hoe sterk de invloed van random is. Uiteindelijk kan een random waarde nooit méér invloed uitoefenen dan bepaald is in de variance.

Ten eerste wordt de stam berekend. Ieder punt wordt telkens op de x- en de z-as minimaal verplaatst. Deze waarde is willekeurig en niet gelijk voor beide assen. Met elk punt wordt de boom een stukje hoger. Hoeveel hoger hangt af van de hoeveelheid segmenten die door de gebruiker bepaald is. Ook deze waarde wordt weer door de random number generator gehaald. De invloed hiervan is minimaal om geen extreme resultaten te krijgen. Als de stam klaar is wordt deze opgeslagen, samen met een aantal handige berekende waarden die later weer van pas komen.

Als dit gebeurt is worden de takken berekend. Dit gebeurt veelal op dezelfde manier als de stam. Een groot verschil is dat ze niet in de oorsprong beginnen. Er moet eerst een punt gekozen worden om vanuit te groeien. Bij de meeste bomen groeien de takken vanaf ongeveer de helft en daarboven. De punten waar ze uit groeien worden door middel van kans berekend. Hoe lager, hoe groter de kans dat daar een tak uit groeit. Een ander ding om rekening mee te houden is dat de takken vanuit de zijkant omhoog afbuigen. Er wordt net zoals bij de stam een punt berekend op basis van de vorige, maar bij de takken wordt de y-waarde van dit punt samen met de x-waarde hoger. Pas als de hele tak genereerd is, wordt deze als geheel in een willekeurige richting gedraaid.



Mesh

Tot slot wordt de mesh eromheen gelegd. Om de logica van meshes te verbergen heb ik dit weggestopt in een apart class. Eerst wordt er voor ieder punt een cirkel omheen getrokken. De straal van deze cirkel is afhankelijk van de hoogte in de boom: hoe hoger, hoe dunner. Volgens een lineair verband was dit niet heel natuurlijk dus heb ik gekozen een wortelverband te gebruiken. Dit zorgt ervoor dat de dikte bovenin sneller afneemt dan onderin.

Hoewel dit aardig werkt voor de stam is het minder geschikt voor de takken. Dit komt omdat de takken zijdelings liggen. Alle cirkels moeten dus gedraaid worden in de richting van de tak. Dit was een technische uitdaging die ik niet heb kunnen oplossen.

Reflectie

Uiteindelijk is het me gelukt om een model voor een redelijk realistisch-uitziende dode boom te maken. Een groot gemis is het bladerdak. De zijtakken van de boom zijn niet erg overtuigend doordat de dikte van de tak op veel plekken niet klopt.

Hoewel deze bomen visueel overtuigend kunnen zijn, klopt er natuurgetrouw niet heel veel van. Ik heb enkele principes van de werkelijkheid kunnen toepassen, namelijk dat takken omhoog groeien of dat deze nooit boven de boom zelf uit zullen komen. Voor een spel maakt dit uiteraard niet veel uit. Iets is een boom als het er op lijkt, en dit voldoet aan dat criterium. Aangezien de stijl van het soort scenes waar dit voor bedoelt is al niet heel realistisch is, maakt het niet uit dat de bomen ook niet natuurgetrouw zijn. Het heeft daarentegen wel een hoop werk gespaard in het uitwerken ervan, maar waarschijnlijk ook in de tijd die het kost om een boom te genereren.