

PELIMOOTTORIEN GRAAFISEN HAHMONNUKSEN VERTAILUA



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus

syksy 2022

Riku Maijala

Tietojenkäsittelyn koulutus

Tekijä Riku Maijala

Työn nimi Pelimoottorien graafisen hahmonnuksen vertailua

Ohjaaja Tommi Lahti

Tiivistelmä

Vuosi 2022

TIIVISTELMÄ

Opinnäytetyön tavoitteena oli vertailla Unreal engine- ja Unity -pelimoottorien hahmonnustyökaluja keskenään ja kertoa uusista ominaisuuksista, mitä on graafiseen hahmonnukseen tulossa.

Opinnäytetyön teoriaosuudessa kerrotaan graafisen hahmonnuksen eri tekniikoista ja tulevista ominaisuuksista. Teoriaosuudessa käydään läpi isoimmat ja tärkeimmät uudistukset, mitä on tulossa Unreal Engine 5.0 versioon. Työssä rakennetaan pelimaailmat käyttäen Unreal engine- ja Unity -pelimoottoria ja kerrotaan lukijalle omat kokemukset maailmojen rakennuksen eri vaiheista.

Vertailussa todettiin, että Unreal Engine 5 -pelimoottorilla pelimaailman luomisen sai näyttämään todella realistiselta vähällä kokemuksella, kun taas Unityllä hahmonnustyökalut olivat samankaltaisia, mutta visuaalinen puoli ei vetänyt vertoja UE5:lle. Johtopäätöksenä todettiin, että Unreal Engine 5 on aloittelevalla pelimaailmojen kehittäjälle helpompi pelimoottori tehdä visuaalisesti näyttävämpiä maailmoja.

Avainsanat Graafinen hahmonnus, Unreal Engine, Unity

Sivut 24 sivua ja liitteitä 1 sivu

Degree Programme in Business Information Technology
Author Riku Maijala
Subject Comparison of graphical rendering of game engines
Supervisors Tommi Lahti

Abstract
Year 2022

ABSTRACT

The aim of the thesis was to compare the rendering tools of the Unreal engine and Unity game engines with each other and to tell about the new features and what is coming to graphic rendering.

The theoretical part of the thesis describes the different techniques and future features of graphic rendering. In the theoretical part, we go through the biggest and most important innovations that are coming to the Unreal Engine 5.0 version. In the practical part, game worlds are built using the Unreal engine and Unity game engine and the reader is told about my experiences of the different stages of building the worlds.

The comparison stated that with Unreal Engine 5, the creation of the game world was made to look very realistic with little experience, while with Unity, the rendering tools were similar, but the visual side did not compare to UE5. The conclusion was that Unreal Engine 5 is an easier game engine for the novice game world developer to make the more visually stunning worlds.

Keywords Graphical rendering, Unreal Engine, Unity

Pages 24 pages and appendices 1 page

Sanasto

Unreal Engine

Pelimoottori

Unity

Pelimoottori

Renderöinti

Graafinen hahmonnus

Emissiivisyys

Kappaleen lähettämän säteilyn määrä. Jos kappaleen heijastavuus on suuri, niin emissiivisyys on pieni

UE5

Unreal Engine 5

CGI

Computer Generated Imagery

Sisällys

1	Johdanto	1
2	Graafinen hahmonnus	2
2.1	Rasterointi	2
2.2	Säteensuuntaus	2
2.3	Säteenseuranta	3
2.4	Polygonit	4
3	Historiaa	5
3.1	1970-luku	5
3.2	1980-luku	6
3.3	1990-luku	7
4	Tulevat Ominaisuudet	8
4.1	Lumen	8
4.1.1	Lumenin globaali valaistus ja heijastus	8
4.1.2	Lumen Säteenseuranta	9
4.2	Nanite virtualisoitu geometria	9
4.2.1	Kuinka Nanite toimii	10
4.2.2	Naniten tuetut ominaisuudet	10
4.3	Virtual shadow maps	11
4.3.1	Shadow Map Ray Tracing	11
5	Demon luonti Unreal Engine	12
5.1	Marketplace ja Quixel	12
5.2	Maapohjan luominen	13
5.3	Valotus	14
5.4	Lopputulos	15
6	Demon luonti Unity	17
6.1	Unity Asset Store	17
6.2	Maapohjan luominen	18
6.3	Valotus	20
6.4	Lopputulos	21
7	Vertailua	23
8	Yhteenveto	24
	Lähteet	25

Kuvat

Kuva 1 Utah Teapot	5
Kuva 2 Dire Straitsin Money For Nothing musiikkivideo	6
Kuva 3 Lumen Final Gather.....	9
Kuva 4 Uuden projektin aloitus.	12
Kuva 5 Quixel.	13
Kuva 6 Quixel-osakappaleita.	14
Kuva 7 Valmis maapohja.....	14
Kuva 8 Directional light.....	15
Kuva 9 UE5 varjostus.	15
Kuva 10 Valmis UE5 maailma.	16
Kuva 11 Tyhjä Unity-projekti.	17
Kuva 12 Unity Asset Store.	18
Kuva 13 Maapohjan luominen.	19
Kuva 14 Asettien lisääminen.....	20
Kuva 15 Unity varjostus.	21
Kuva 16 Unity sumu.....	22
Kuva 17 Unity lopputulos.	22

Liitteet

Liite 1	Aineistonhallintasuunnitelma
---------	------------------------------

1 Johdanto

Opinnäytetyön tavoitteena on tuoda lukijalle laajalti tietoa graafisesta hahmonnuksesta eli renderöinnistä. Opinnäytetyössäni tutkin Unity- ja Unreal Engine -pelimoottorien graafisia hahmonnustyökaluja ja vertailen niitä keskenään. Työssä keskitytään etsimään uutta tietoa, mitä on tulossa renderöintiin.

Työssä etsin tietoa renderöinnistä, kerron sen historiasta ja tulevista ominaisuuksista. Vertailen saatavilla olevan tiedon määrää maailmojen rakentamisessa kummallakin pelimoottorilla opinnäytetyön teon hetkellä. Molemmilla pelimoottoreilla luodaan demo ja testillaan eroavaisuuksia näiden välillä.

Tämän opinnäytetyön ensisijainen tehtävä on auttaa peliohjelmoinnista kiinnostunutta, jolla ei ole aikaisempaa kokemusta maailmojen luomisesta graafisen hahmonnuksen pariin.

Tutkimuskysymykseni olivat:

- Mitä graafisen hahmonnuksen uudet ominaisuudet tuovat pelien kehittämiseen?
- Miten Unreal Enginen hahmonnustyökalut eroavat Unityn hahmonnustyökaluista?
- Kuinka saada aloitteleva pelinkehittäjä hahmonnuksen pariin?

2 Graafinen hahmonnus

Graafinen hahmonnus, joka tunnetaan myös nimellä renderöinti, on toimenpide, jossa käytetään tietokoneohjelmistoa fotorealisticen tai ei-fotorealisticen kuvan luomiseksi 2D- tai 3D-mallista. Renderöinti on lopulliselle kuvalle annettu nimi. Kohtaustiedostolla, joka sisältää objekteja tarkasti määritellyllä kielellä tai tietorakenteella, voi olla useita malleja. Kohtaustiedosto sisältää tietoja virtuaalisen kohtauksen geometriasta, näkökulmasta, tekstuurista, valaistuksesta ja varjostuksesta. Kohtaustiedoston tiedot toimitetaan sitten hahmonnusohjelmaan, joka käsittelee ne ja tulostaa sen digitaalisena kuvana tai rasterigrafiikkakuvatiedostona. Hahmonnusmenetelmiin kuuluvat muun muassa rasterointi, säteensuuntaus ja säteenseuranta. (Wikipedia, 2022)

2.1 Rasterointi

Rasterointi on yksi tyypillisistä 3D-mallien hahmonnustekniikoista. Verrattuna muihin hahmonnustekniikoihin, kuten säteenseurantaan, rasterointi on erittäin nopeaa ja siksi sitä käytetään useimmissa reaaliaikaisissa 3D-koneissa. Rasterointi on prosessi, jossa lasketaan kartoitus kohtauksen geometriasta pikseleihin. Jokaisen pikselin tietty väri määräytyy pikselin varjostimella. Varjostuksessa voidaan ottaa huomioon fyysisiä vaikutuksia, kuten valon asentoa, niiden likiarvoja tai puhtaasti taiteellista tarkoitusta.

Rasteroinnin laatua voidaan parantaa antialiasoinnilla, joka luo "sileät" reunat. Osapikselin tarkkuus on menetelmä, joka ottaa huomioon paikat tarkemmassa mittakaavassa kuin pikseliruudukko ja voi tuottaa erilaisia tuloksia, vaikka primitiivin päätepisteet osuvat samoihin pikselikoordinaatteihin, mikä tuottaa sujuvampia liikeanimaatioita. (Computer Hope, 2017; Wikipedia)

2.2 Säteensuuntaus

Säteensuuntauksessa mallinnettua geometriaa jäsennetään pikseli pikseliltä, rivi riviltä, näkökulmasta ulospäin. Kohteessa, jossa objekti on leikattu, väriarvo pisteessä voidaan arvioida useilla menetelmillä. Yksinkertaisimmillaan kohteen väriarvosta leikkauspisteessä tulee kyseisen

pikselin arvo. Väri voidaan määrittää tekstuurikartalta. Kehittyneempi menetelmä on muuttaa väriarvoa valaistuskertoimella, mutta ilman suhdetta simuloituun valonlähteeseen. Artefaktien vähentämiseksi voidaan laskea keskiarvo useista säteistä hieman eri suuntiin.

Säteensuuntaus sisältää näkymäsuunnan laskemisen kameran asennosta ja sen säteen heiton asteittaisen seuraamisen kohtauksen kiinteiden 3D-objektien läpi, samalla kun kerätään tuloksena oleva arvo jokaisesta 3D-avaruuden pisteestä. Säteensuuntaus tarkoittaa, että valonsäde seuraa suoraa polkua, johon voi kuulua kulkeminen puoliksi läpinäkyvien esineiden läpi. Säteensuuntaus on vektori, joka voi olla peräisin kamerasta tai kohtauksen päätepisteestä. Joskus lopullinen valoarvo johdetaan siirtofunktiosta ja joskus sitä käytetään suoraan.

Voidaan myös käyttää optisten ominaisuuksien karkeita simulaatioita: suoritetaan yksinkertainen säteen laskenta kohteesta näkökulmaan. Toinen laskelma tehdään valonlähteestä tulevan valonsäteiden tulokulmasta ja näistä sekä valonlähteiden määritellyistä intensiteetistä lasketaan pikselin arvo. Toinen simulaatio käyttää valaistusta, joka on piirretty radiositeettialgoritmilla tai näiden kahden yhdistelmällä. (Wikipedia, 2022; Computer Hope, 2019)

2.3 Säteenseuranta

Säteenseuranta on tekniikka, jolla simuloidaan valon luonnollista virtausta hiukkasina. Kyseessä on valaistuksen simulointi tietokoneohjelmalla. Säteenseurannan keskeinen menetelmä on jäljittää valonlähteistä lähtevien säteiden törmäyksiä kappaleisiin ja edelleen kameran linssiin.

Valonlähteiden säteiden törmäykset esineisiin ja kauemmas kameran linssiin ovat säteenseurannan keskeisin menetelmä. Säteenseuranta pystyy simuloimaan erilaisia optisia tehosteita kuten heijastuksia, taittumista, pehmeitä varjoja, sirontaa, syväterävyyttä, liikkeen epäterävyyttä, kaustiikkaa, ympäristön okklusio- ja dispersioilmiöitä. (Wikipedia, 2022; Computer Hope, 2021)

2.4 Polygonit

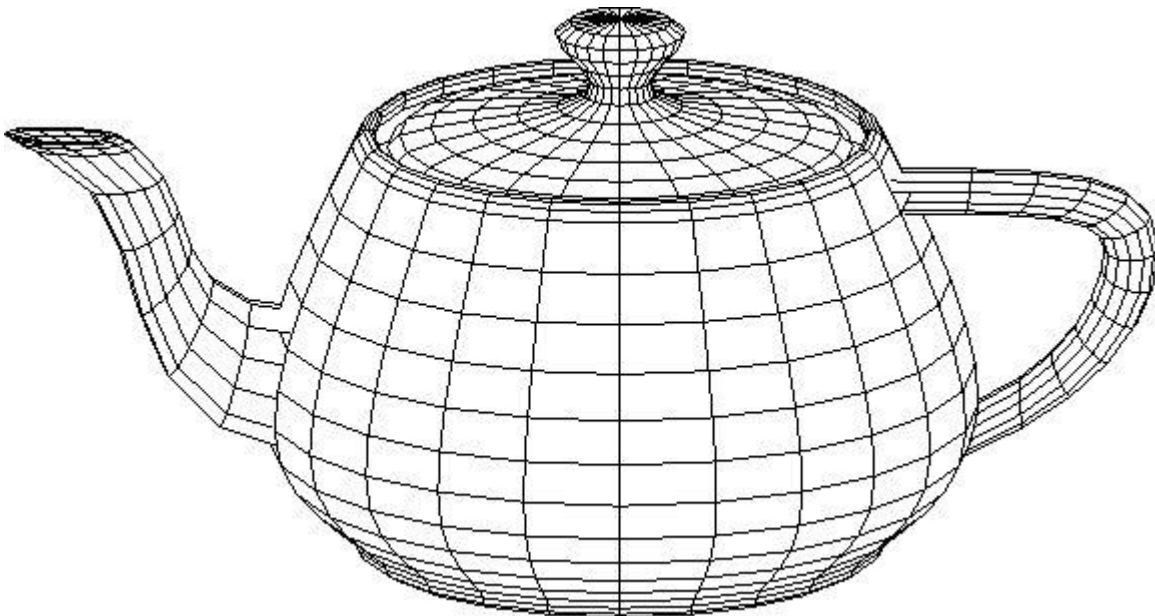
Polygoni on kolmiulotteisessa grafiikassa renderöintiin käytettävä monikulmion kantainen primitiivi. Renderöinti ei salli polygoneja, joissa ääriviivat leikkaavat toisiaan, jotka ovat koveria tai joissa on reikiä. Yleensä kolmion muotoiset polygonit syntyvät, kun objektin pinta mallinnetaan, kärkipisteet valitaan ja objekti renderöidään lankakehysmallissa. Mitä enemmän polygoneja esiintyy kohtauksessa, sitä kauemmin sen renderöinti kestää.

3 Historiaa

3.1 1970-luku

Tietokonegrafiikan tutkimus ulottuu 1960-luvulle asti, jolloin tietojenkäsittelytieteilijä ja professori Ivan Sutherland esitteli Sketchpad-nimisen ohjelmiston, mutta vasta 1970-luvulla tutkijat alkoivat ottaa suuria harppauksia alalla. Koko 1970-luvun Sutherland ja hänen kollegansa David C. Evans olivat edelleen 3D-renderöinnin ja grafiikan kärjessä opettaen Utahin yliopistossa ja Utah osoittautui 3D-renderöinnin keskuksiksi. Vaikka Sutherlandin Sketchpad-ohjelma pystyi tuottamaan vain alkeellisimmat 3D-muodot, muut tietotekniikan tutkijat käyttivät tätä työkalua hyppykohtana. 1970-luvun lopulla he pystyivät renderöimään monimutkaisia muotoja 3D-muodossa, joista yksi tunnetuimmista oli "Utah Teapot" (Kuva 1), jonka ensimmäisenä renderöi Martin Newell Utahin yliopistossa. Teekannun monien käyrien ja ääriviivojen toistamiseen tehdyt laskelmat edustivat suurta harppausta eteenpäin 3D-renderöinnissä.

Kuva 1 Utah Teapot



Yksi Sunderlandin oppilaista Edwin Catmull, loi ensimmäisen 3D-animoidun mallin käyttämällä Z-puskurointia. Hänen vuotensa 1972 opiskelijaelokuva "A Computer Animated Hand" oli yksi varhaisimmista esimerkeistä animoidusta CGI:stä.

3.2 1980-luku

1980-luvulle siirtyessä tietojenkäsittelytieteilijät ja liikemiehet ymmärsivät nopeasti 3D-tietokonegrafiikan mahdollisuudet. 1980-luvun puoliväliin mennessä pintakuviokartoitus mahdollisti ammattilaisten, kuten arkkitehtien, luoda 3D-renderöityjä malleja luomuksistaan. 3D-grafiikka alkoi myös tunkeutua populaarikulttuuriin. Sitä esitettiin elokuvissa ja musiikkivideoissa, kuten esimerkiksi Dire Straitsin Money For Nothing-musiikkivideolla (Kuva 2).

Kuva 2 Dire Straitsin Money For Nothing musiikkivideo



Ensimmäinen elokuva, jossa hyödynnettiin laajasti kiinteää 3D-CGI:tä, oli Walt Disneyn Tron vuonna 1982. Elokuvaa käytettiin animaatioaikaa alle 20 minuuttia. Kohtausten luomiseksi Disney kääntyi neljän johtavan tietokonegrafiikkayrityksen puoleen: Information International Inc, Robert Abel and Associates, MAGI ja Digital Effects.

3.3 1990-luku

1990-luvulla 3D-renderöintitekniikka kehittyi nopeammin kuin missään muussa vaiheessa historiaansa. Vuonna 1995 saapui elokuva Toy Story, joka oli ensimmäinen kerta, kun elokuva esitettiin täydessä 3D-grafiikassa. Myös videopelit kehittyivät nopeasti tänä aikana ja 2D-pikselitaiteesta siirryttiin nopeasti täysin renderöityyn 3D-grafiikkaan.

Autodesk julkaisi ensimmäisen 3D-studion vuonna 1990, jonka takana olivat Tom Hudson ja Dan Silva. Ensimmäinen versio soveltui spline-viivojen, loft-pintojen, geometrysten primitiivien/polygonien ja perusverkkomuokkauksen piirtämiseen.

Vuonna 2000 Paul Debevecin johtama ryhmä onnistui vangitsemaan ja simuloimaan riittävän hyvin ihmisen kasvojen heijastuskentän käyttämällä yksinkertaisimpia valoasteita, joka oli viimeinen puuttuva pala tehdä näyttelijöistä digitaalisia kopioita.

4 Tulevat Ominaisuudet

Viimeisimpien Unreal Engine 5:n hahmonnusteknologioiden avulla voit luoda avoimen maailman pelejä, jotka hyödyntävät reaaliaikaista globaalia valaistusta, heijastuksia ja korkearesoluutioista varjostusta. Näiden parannusten avulla voit luoda seuraavan sukupolven mukaansatempaavia ja realistisia pelikokemuksia.

4.1 Lumen

Lumen on uusi täysin dynaaminen globaali valaistus- ja heijastustekniikka Unreal Engine 5:ssä, joka on optimoitu seuraavan sukupolven konsoleille. Lumen on Unreal Engine 5:n oletusarvoinen globaali valaistus- ja heijastusjärjestelmä. Millimetreistä kilometreihin vaihtelevalla etäisyydellä se heijastaa hajanaisia heijastuksia äärettömillä pomppauksilla ja epäsuorilla heijastuksilla laajoissa, yksityiskohtaisissa kohtauksissa. (Unreal Engine 5, 2022)

4.1.1 Lumenin globaali valaistus ja heijastus

Lumen ratkaisee dynaamisen hajavalon epäsuoran valaistuksen. Värivuotoa tapahtuu, kun pinnasta hajanaisesti pomppiva valo poimii pinnan värin ja heijastaa värillisen valon muille lähellä oleville pinnoille. Epäsuoraa valaistusta estävät myös kohtauksen verkot, mikä johtaa epäsuoraan varjostukseen.

Final Gather Quality -ominaisuus parantaa Lumenin valaistuksen laatua ja vähentää renderöityä kohinaa, mutta lisää grafiikkasuorittimen kustannuksia renderöimisestä.

Lumenin Final Gather levittää valoa emissiivisten materiaalien läpi ilman ylimääräistä suorituskykyä. Taivaan valaistus on ratkaistu osana Lumenin Final Gatheriä (kuva 3). Tämä sisältää taivaanvarjostuksen, joka mahdollistaa sisätilojen olevan paljon tummempia kuin ulkona. Emissiivisten tilojen kokoa ja kirkkautta on kuitenkin rajoitettava, ennen kuin ne aiheuttavat kohina-arteefakteja. Tämä on luonnostaan vaikeampi korjata kuin manuaalisesti sijoitettavat valonlähteet. (Unreal Engine 5, 2022)

Kuva 3 Lumen Final Gather.



Lumen käyttää oletusarvoisesti ohjelmistosäteenseurantaa allekirjoitettujen etäisyyskenttien kautta, mutta se voi saavuttaa korkeamman laadun tuetuilla näytönohjaimilla, kun laitteistosäteenseuranta on käytössä.

4.1.2 Lumen Säteenseuranta

Lumen käyttää oletusarvoisesti ohjelmistosäteenseurantaa allekirjoitettujen etäisyyskenttien kautta, mutta se voi saavuttaa korkeamman laadun tuetuilla näytönohjaimilla, kun laitteistosäteenseuranta on käytössä.

4.2 Nanite virtualisoitu geometria

Nanite on Unreal Engine 5:n uusi virtualisoitu geometriajärjestelmä, joka renderöi pikseliasteikon yksityiskohdat ja korkean objektimäärän käyttämällä uutta sisäistä verkkomuotoa ja renderöintimoottoria. Se käsittelee älykkäästi vain havaittavissa olevia yksityiskohtia, ei enempää.

Naniten tietomuoto on myös erittäin pakattu, ja se tukee hienojakoista suoratoistoa automaattisella tarkkuudella. (Unreal Engine 5, 2022)

4.2.1 Kuinka Nanite toimii

Naniittiverkko on pohjimmiltaan kolmioverkko. Nanite käyttää täysin uutta järjestelmää tietomuodon hahmontamiseen tehokkaalla tavalla. Se pystyy käsittelemään suuruusluokkaa enemmän kolmioita ja esiintymiä, kuin on mahdollista perinteisesti renderöidyssä geometriassa. Materiaalit on kohdistettu verkon osiin siten, että niissä voidaan käyttää erilaisia varjostusmalleja ja dynaamisia tehosteita. Virtuaalikuvioita ei vaadita käyttämään Naniten kanssa, mutta ne ovat erittäin suositeltavia.

Nanite integroituu mahdollisimman saumattomasti olemassa oleviin moottorien työkulkuihin samalla kun se käyttää uutta lähestymistapaa verkkotietojen tallentamiseen ja hahmontamiseen. Verkot analysoidaan ja jaetaan kolmioryhmien hierarkkisiin klustereihin tuonnin aikana. Renderöinnin aikana klusterit vaihdetaan lennossa eri tarkkuudella kameranäkymän mukaan. Nanite toimii omassa renderöintipassissaan, joka ohittaa täysin perinteiset piirtokutsut. Visualisointitiloja voidaan käyttää Nanite-putkilinjan tarkastamiseen. (Unreal Engine 5, 2022)

4.2.2 Naniten tuetut ominaisuudet

Naniteverkkoa voidaan käyttää seuraavien komponenttityyppien kanssa: staattinen verkko, esiintymä staattinen verkko, Hierarkkinen esiintymä staattinen verkko ja geometria kokoelma. Nanite tukee dynaamista siirtymistä, kiertoa ja epätasaista jäykkien verkkojen skaalausta, mutta ei tue yleistä verkon muodonmuutosta. Kohtauksessa olevien esiintymien enimmäismäärä on rajoitettu kahteen miljoonaan esiintymään. Tämä sisältää kaikki esiintymät, jotka on suoratoistettu. (Unreal Engine 5, 2022)

4.3 Virtual shadow maps

Virtual Shadow Maps (VSM) on uusi varjokartoitusmenetelmä, jota käytetään tuottamaan johdonmukainen, korkearesoluutioinen varjostus, joka toimii elokuvaalaatuisten resurssien ja suurten, dynaamisesti valaistujen avoimien maailmojen kanssa käyttämällä Unreal Engine 5:n Nanite virtualisoitua geometriaa, Lumen globaali valaistus ja heijastus -tekniikkaa ja Maailmanosio -ominaisuuksia. Se on suunniteltu lisäämään merkittävästi varjojen resoluutiota vastaamaan erittäin yksityiskohtaista naniittigeometriaa. Se myös korvaa monet kiinteän valon varjostustekniikat yhdellä, yhtenäisellä polulla. (Unreal Engine 5, 2022)

4.3.1 Shadow Map Ray Tracing

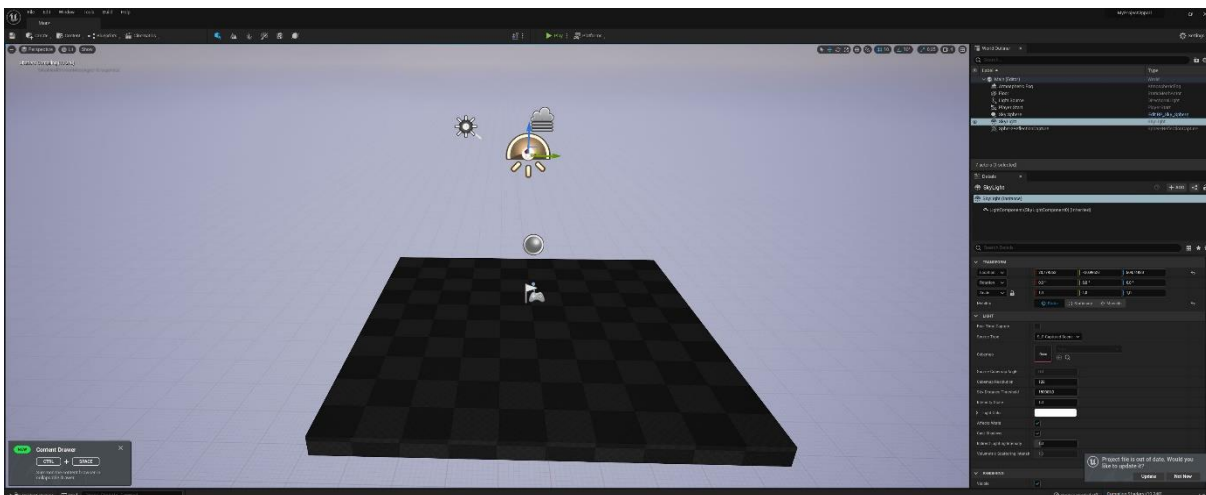
Shadow Map Ray Tracing on näytteenottoalgoritmi, jota käytetään virtuaalisten varjokarttojen kanssa tuottamaan pehmeitä varjoja. Objektit, jotka luovat varjoja kauemmaksi, saavat pehmeämpiä varjoja kuin esineet, jotka luovat varjoja lähemmäksi varjoja vastaanottavaa pintaa. Algoritmi toimii ampumalla useita säteitä kohti valonlähdettä, mutta sen sijaan, että se arvioisi leikkauskohtia geometrian kanssa kuten perinteinen säteenseuranta tekee, useita säteen varrella olevia näytteitä esitetään ja testataan virtuaalista varjokarttaa vasten pehmeän varjon saavuttamiseksi.

Shadow Map Ray Tracing laskee keskimääräisen valon pikseliä kohden. Mitä enemmän valoja koskettaa suuria osia näytöstä, sitä kalliimpaa renderöinti on. Valot, jotka kattavat pienen määrän näytöllä olevia pikseleitä, ovat halvempia. Varjojen näkyvä pehmeys vaikuttaa suorituskykyyn mukautuvan sädemäärän ansiosta. (Unreal Engine 5, 2022)

5 Demon luonti Unreal Engine

Demon aloitus alkaa erittäin simppeillä näkymällä uuden projektin aloittaessa (kuva 4). Projektista poistettiin valmis alusta ja pelaajan sijainti.

Kuva 4 Uuden projektin aloitus.



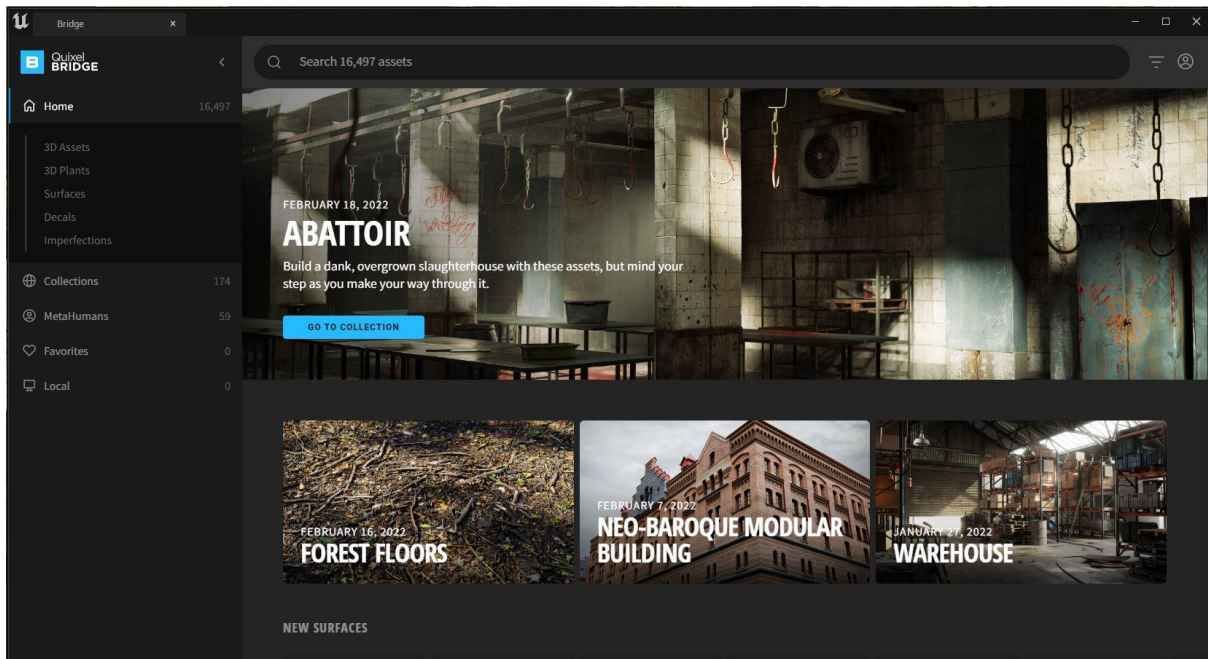
5.1 Marketplace ja Quixel

Unreal Engineä varten tarvitsee ladata Epic Games Launcher, jossa on valmiina oma Marketplace, mistä saa ladattua valmiita paketteja maailmojen rakentamista varten. Latasin omaan projektiin Iceland Starting point -nimisen paketin, josta sai helposti upotettua valmiiksi ladattuja asetteja projektiin. UE5 -moottorilla pääsi myös helposti Quixel-nimiseen sovellukseen, mistä sai ladattua helposti yksittäisiä asetteja.

Quixelin perusti Teddy Bergsman ja Waqar Azim vuonna 2011 ja se liittyi osaksi Epic Gamesia vuonna 2019. Quixel käyttää yhtä maailman suurimmista fotogrammerikirjastoista ja mukana toimitetuista kirjastoista: Megascans, Bridge ja Mixer. Megascans on valtava online-skannauskirjasto, joka sisältää korkearesoluutioisia, johdonmukaisia PBR-kalibroituja pinta-, kasvillisuus- ja 3D-skannauksia, myös työpöytäsovelluksien ladattujen skannaustietojen hallintaan, sekoittamiseen ja vientiin. Bridge (Kuva 5) on UE5-pelimootorin laajennus, joka antaa Quixelin

kirjaston käyttöön pelimoottorille. Mixer on helppokäyttöinen 3D-kuviointisovellus, johon kuuluu ilmaisia Smart materials- ja Megascans-resursseja, jotka voivat nopeuttaa teksturointiprosessia.

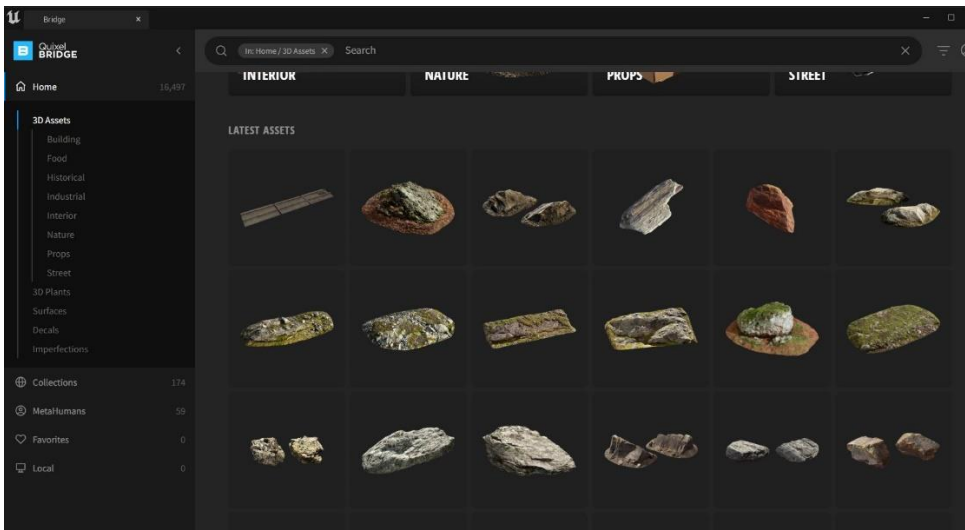
Kuva 5 Quixel.



5.2 Maapohjan luominen

Maan luominen alkoi ottamalla yksi osakappale (Kuva 6) kerrallaan ja niitä muotoiltiin ja pyöriteltiin testimielessä ympäriinsä. Kun pari samanlaista osakappaletta oli yhdistelty ja niistä tehty isompi kokonaisuus, oli helppo alkaa monistaa isompaa kokonaisuutta ja jatkaa alueen kasvattamista (Kuva 7).

Kuva 6 Quixel-osakappaleita.



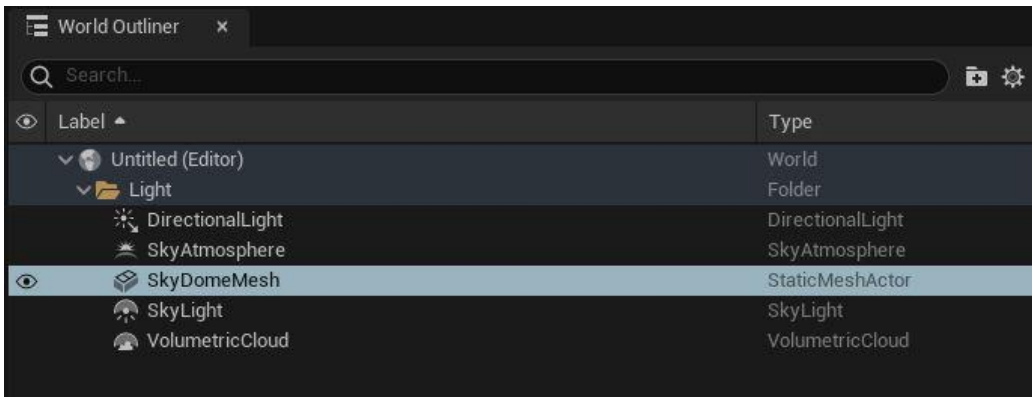
Kuva 7 Valmis maapohja.



5.3 Valotus

Environment Light Mixer on telakoitava muokkausikkuna, jossa voi lisätä muokata taivaan, pilvien, tunnelmavalojen ja taivaanvalon ympäristövalaistuskomponenttien ominaisuuksia. Directional light (Kuva 8) oli auringon paikka ja tämän paikkaa pystyi helposti vaihtamaan painamalla CTRL + L -näppäimiä. Aurinkoa siirtämällä sai helposti siirrettyä varjostuksia demossa omaan haluamaansa suuntaan. Ilmakehää sai siirrettyä joko ylös tai alas ja tällä sai myös tehtyä erilaisia varjostuksia demoon.

Kuva 8 Directional light.



Varjot UE5 editorissa näyttivät realistisilta ja hyvin tarkoilta (kuva 9).

Kuva 9 UE5 varjostus.



5.4 Lopputulos

Demon luominen UE5:lla alkoi nopeasti ja tietoa oli saatavilla useista paikoista, kuinka editoria käytetään. Editorissa on paljon helposti käytettäviä työkaluja, kuinka kappaleita esimerkiksi muokataan ja monistetaan. Lisäsin omaan projektiini sumua kirjoittamalla editorissa olevaan hakukenttään sanan: fog. Tätä kautta pääsin suoraan visuaalisiin efekteihin, jossa pystyi muokkaamaan esimerkiksi sumua, heijastuksia, taivaan tunnelmaa ja jälkikäsitteilyn määrää.

Sumun väriä ja läpinäkyvyyttä pystyi helposti muokkaamaan ja tämä toi omaan testimaailmaani enemmän reaalisuutta (Kuva 10).

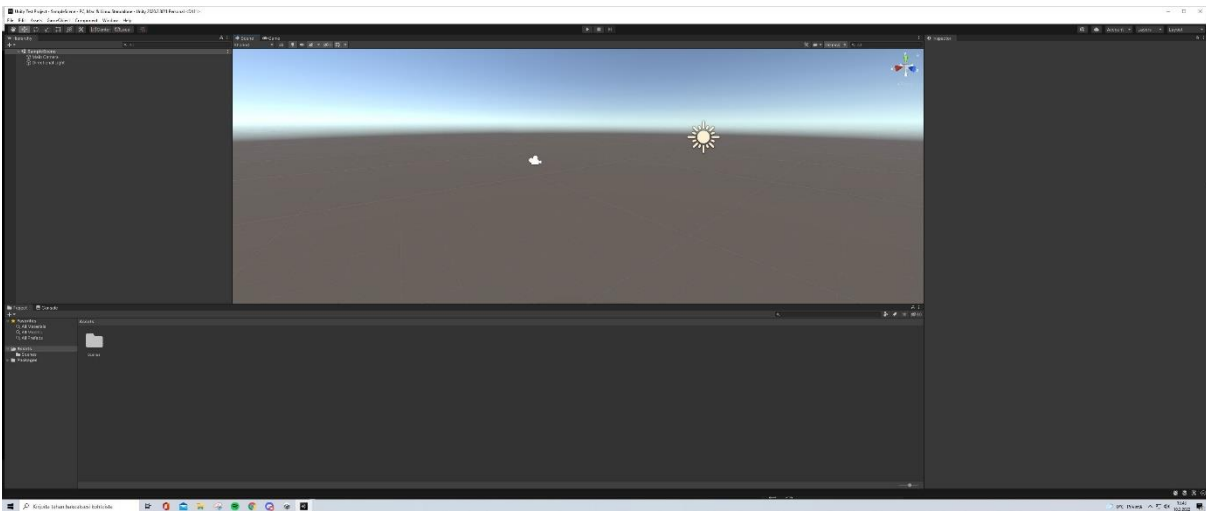
Kuva 10 Valmis UE5 maailma.



6 Demon luonti Unity

Demon luominen Unity -pelimoottorilla alkoi hyvin samankaltaisesti kuin UE5:lla. Aurinko ja taivas oli valmiiksi paikoillaan muuten tyhjässä olevassa projektissa. (kuva 11).

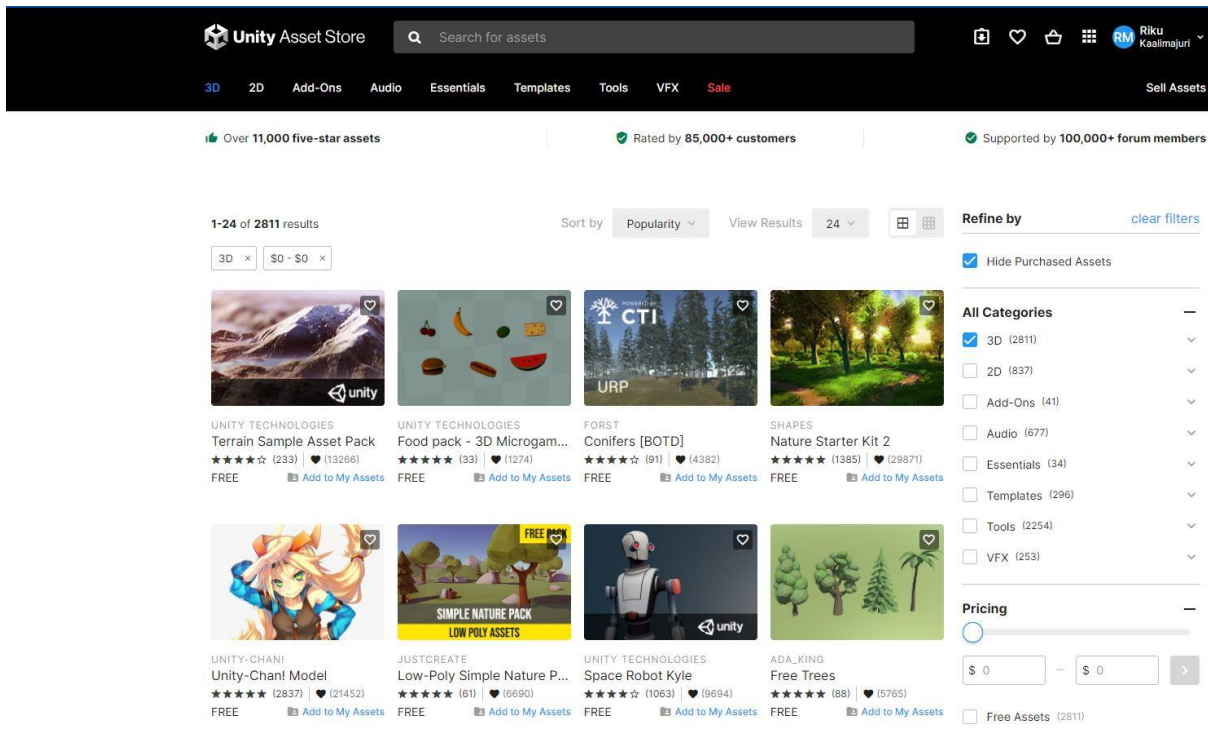
Kuva 11 Tyhjä Unity-projekti.



6.1 Unity Asset Store

Unity Asset Storeen (Kuva 12) pääsee joko verkkoselaimella kirjoittamalla hakukenttään <https://assetstore.unity.com/> tai avaamalla sen Unityn editorista. Asettien lataaminen vaatii, että kirjaudut sivulle. Latasin NatureStarterKit2 asetin omaan projektiini, sillä se oli ilmaiseksi ladattava ja suurin osa aseteista oli maksullisia. Asetin ladattua avautuu nappula, jota kautta saa suoraa asennettua sen meneillään olevaan projektiin.

Kuva 12 Unity Asset Store.

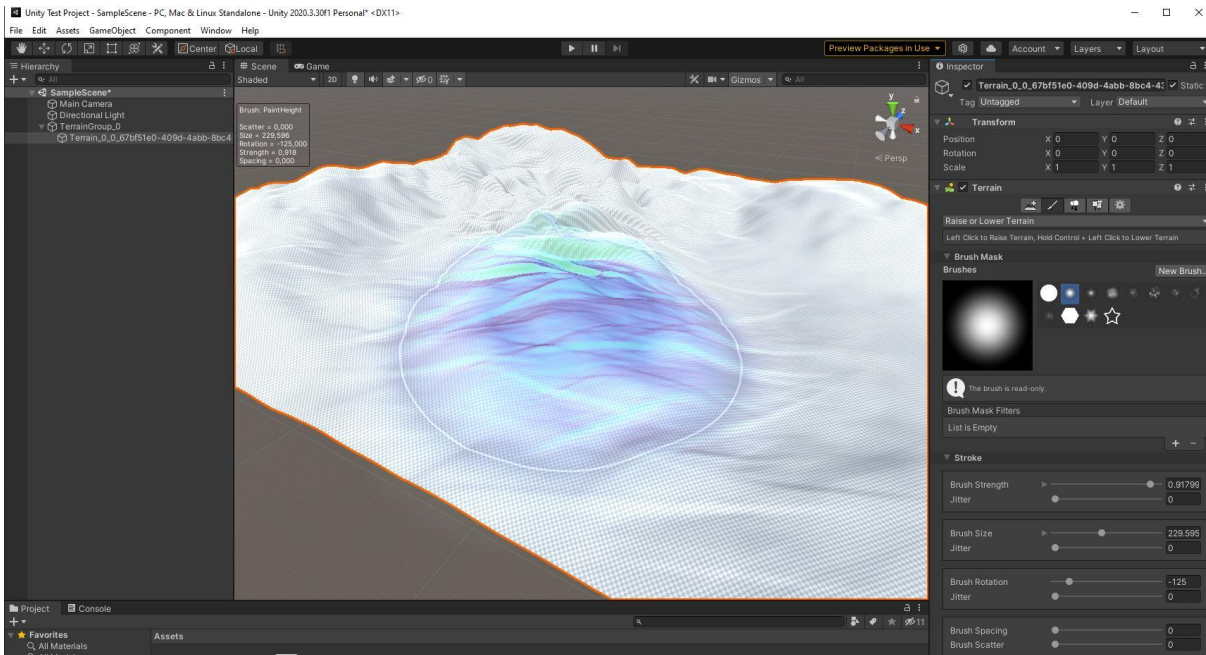


6.2 Maapohjan luominen

Maapohjan luominen alkoi laittamalla yksinkertaisesti maapohja ylävalikoista projektiin.

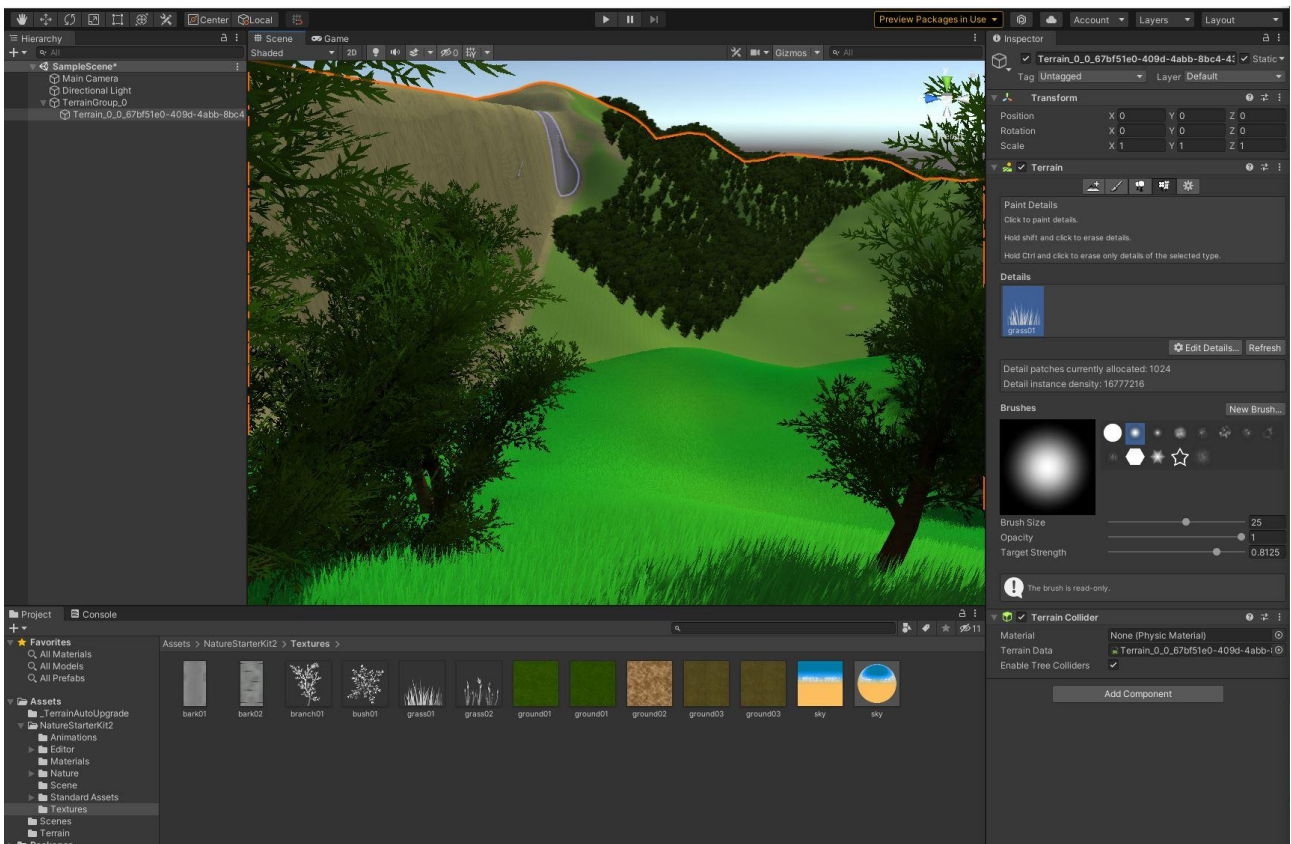
Maapohjaa pystyi muokkaamaan horisontaalisesti muutamalla eri työkalulla, mitä kuvassa (kuva 13) oikeassa reunassa on. Näillä työkaluilla pystyi helposti muokkaamaan maapohjasta vuoristomaisen, mutta muuta järkevää käyttöä näille työkaluille en itse huomannut.

Kuva 13 Maapohjan luominen.



Kun olin saanut vuoristomaisen maiseman tehtyä, oli aika lisäillä asetteja ladatusta paketista ja värjätä maapohja oman maun mukaan (kuva 14).

Kuva 14 Assettien lisääminen.



6.3 Valotus

Unityssä oli hyvin samankaltaiset valotusasetukset kuin Unrealissa. Directional Light ominaisuuden käyttöön ja auringon suunnan pystyin päättämään mistä päin valo heijastui. Unityn varjostukset näyttivät heikkolaatuisilta ja editorin oli oltava lähellä varjostusten renderöimiseksi (kuva 15).

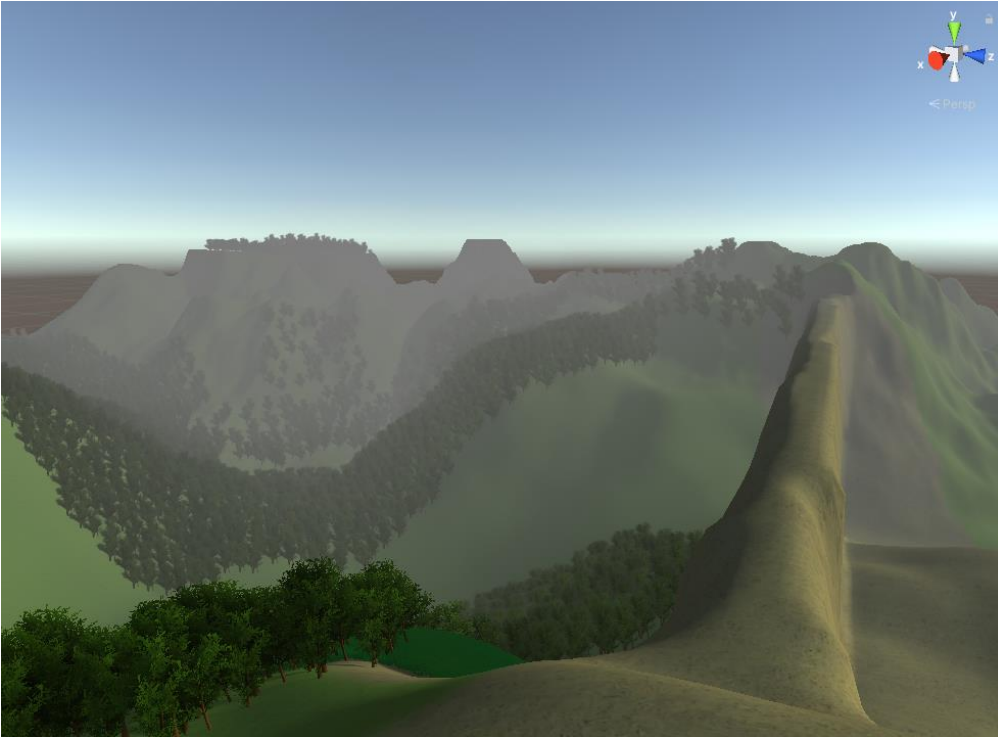
Kuva 15 Unity varjostus.



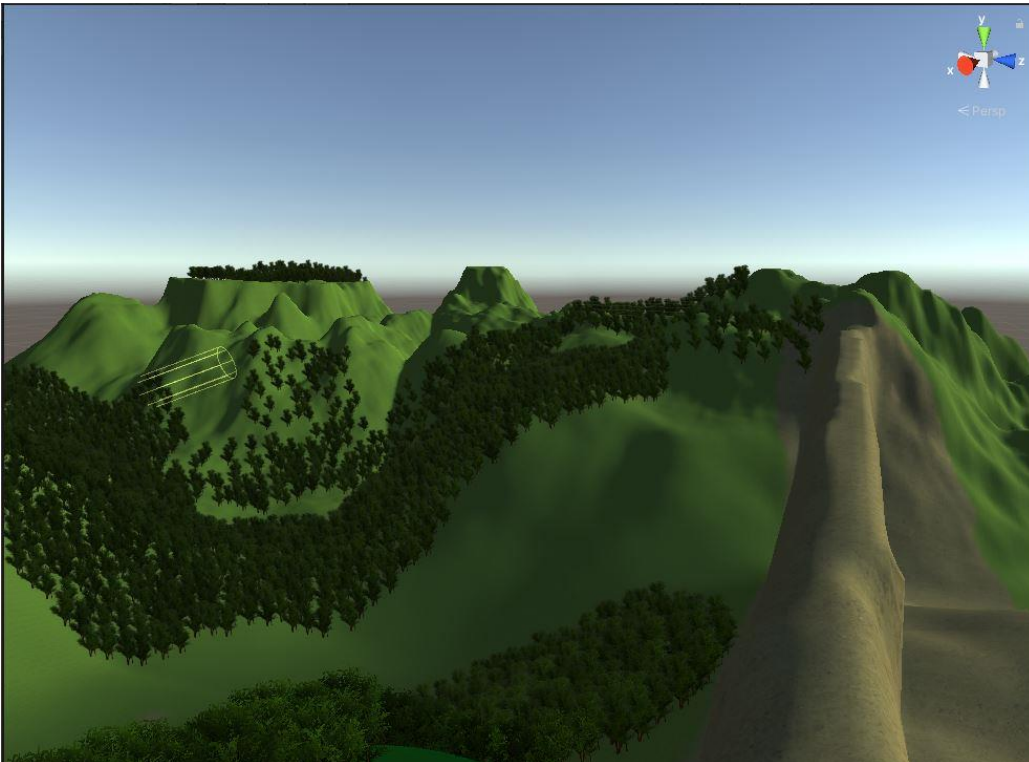
6.4 Lopputulos

Maailman rakentaminen alkoi hitaasti, sillä maaperän muokkaamiseen tarkoitettut työkalut olivat herkkiä liikkeelle ja tästä syystä haluaman mallin luominen saattoi kestää välillä hyvinkin pitkään. Editorilla oli myös mahdollista lisätä sumua maailmaan, mutta se ei näyttänyt vakuuttavalta (kuva 16). Asettien lisääminen onnistui helposti, mutta niiden muokkaamiselle ei ollut mahdollisuutta, joten maailma alkoi näyttämään itseään toistavalta (Kuva 17).

Kuva 16 Unity sumu



Kuva 17 Unity lopputulos.



7 Vertailua

Unity- ja Unreal Engine -pelimoottorit molemmat tukevat 2D- ja 3D-animointia graafisessa hahmonnuksessa. Fyysisesti perustuva renderöinti, globaali valaistus, edistyneet varjostimet, volumetriset valot ja jälkikäsitteily ovat yhteisiä ominaisuuksia molemmille pelimoottoreille.

Unityyn verrattuna UE5 pystyy luomaan tarkempaa ja tasaisempaa valaistusta. Visuaalisia efektejä oli Unrealilla enemmän tarjolla ja tarkemmat varjot tekivät renderöinnistä realistisemmän näköistä.

Unreal Engine 5:ssä oli sisäänrakennettu AI-käyttäytymispuu, mitä ei Unityssä ollut ollenkaan. Tätä varten kyseisen ominaisuuden joutuisi ostamaan Unity Asset Storesta, jos sellaisen haluaisi projektiinsa.

Quixel helpotti erittäin paljon UE5 maailman luomisessa, sillä kerran kirjautumalla sisään pääsi käsiksi kaikkiin tarjolla oleviin asetteihin ilmaiseksi. Unity Asset Storesta oli erilaisia asetteja yli 44 000 kappaletta, mutta niistä ilmaisia oli vain hieman alle 3000 kappaletta.

Unityllä puolestaan on isompi yhteisö, joten kysymyksiä ja vastauksia on enemmän tarjolla. 2D-pelien kehittämisessä moni suosii myös Unityä enemmän helppokäyttöisyyden ja laajemman tuen vuoksi.

Molempiin demojen luomiseen Unityllä ja UE5:llä käytin yhtä paljon aikaa ilman minkäänlaista kokemusta kummastakaan pelimoottorista. UE5 demon avuksi löytyi helposti ajankohtaisia tutoriaaleja ja neuvoja. Unityn avuksi löytyi vähemmän materiaalia 3D-maailmojen rakennuksessa, ja osa editorin asetuksista oli muuttunut, mistä tietoa löytyi.

8 Yhteenveto

Unreal Engine 5:en tulevat ominaisuudet saavat maailmojen luomisen näyttämään todella realistiselta ja elokuvamaiselta. Pelimoottoriin on helppo ladata jo valmiiksi tehtyjä paketteja, joista saa helposti rakennettua haluamaasi teemaan sijoittuvan maailman. Mikäli etsimääsi kappaletta ei löydy ladatusta paketistasi, se voidaan helposti etsiä ja ladata Epic Gamesin sisäisestä kauppapaikasta, joka on liitettyä itse pelimoottoriin. Unityn pelimoottorissa oli joitain samoja ominaisuuksia, kuin UE5:ssa, mutta maailmojen luominen ei näyttänyt yhtä realistiselta.

Etsiessäni tietoa pelimoottoreista, huomasin että suurin osa Unityn maailmanrakennuksesta sijoittui vuodelle 2017 ja sitä ennen. Unreal Engine 5:lle löytyi ohjeita ja tutoriaaleja, jotka olivat ajankohtaisia. Unityä suosivat monet ohjelmoijat tänäkin päivänä sen helppokäyttöisyyden vuoksi koodaamisessa, mutta maailmojen rakentamiseen UE5 on näyttävämpi artisteille ja suunnittelijoille.

Unreal Enginen Nanite-teknologia on isoimpia muutoksia mitä pelien kehittämiseen on tulossa. Se mahdollistaa täysilaatuisen elokuvamateriaalin renderöinnin reaaliajassa. Tämä tarkoittaa, että artistit voivat luoda resursseja, joissa on kymmeniä miljoonia polygoneja, jotka skaalautuvat miljardeihin. Aiemmissa Unreal Engine -malleissa massiivisten geometrinen yksityiskohtien omaavien esineiden sijoittaminen kohtauksen keskelle saattoi johtaa voimakkaaseen hidastumiseen. Nanite avaa mahdollisuuden käyttää erittäin yksityiskohtaista geometriaa säilyttäen samalla hyvän suorituskyvyn.

Lopputulokseni on se, että aloittelevaa pelinkehittäjää suosin maailmojen rakentamisessa ja visuaaliseen puoleen keskittyessä opettelemaan käyttämään Unreal Engine 5 -pelimoottoria. Uudet graafiseen puoleen keskittyvät ominaisuudet, mitä on Unreal Engine 5:lle tulossa saavat aloittelevankin pelinkehittäjän rakentamaan maailmoja, jotka näyttävät henkeäsalpaavilta.

Lähteet

Rendering. (2022). Wikipedia verkkosivusto.

[https://en.wikipedia.org/wiki/Rendering_\(computer_graphics\)](https://en.wikipedia.org/wiki/Rendering_(computer_graphics))

Rasterisation. (2022). Wikipedia verkkosivusto. <https://en.wikipedia.org/wiki/Rasterisation>

Computer Hope. (2017). Rasterize. <https://www.computerhope.com/jargon/r/rasterize.htm>

Computer Hope. (2019). Ray casting. <https://www.computerhope.com/jargon/r/ray-casting.htm>

Computer Hope. (2021). Ray tracing. <https://www.computerhope.com/jargon/r/ray-tracing.htm>

Unreal Engine 5.0 Release Notes. (2022). *Overview of new and updated features in unreal 5.0*

<https://docs.unrealengine.com/5.0/en-US/unreal-engine-5-0-release-notes/>

Unity rendering and visual effects roadmap. (2022). <https://unity.com/roadmap/unity-platform/rendering-visual-effects>

First rendering. <https://firstrender.net/>

History of 3D Rendering <https://pixelperfect-studios.com/history-of-3d-rendering/>

Liite 1: Aineistonhallintasuunnitelma

Opinnäytetyötä varten luotiin Teams-palvelimelle varakopio opinnäytetyöstä mahdollisen tietokoneen rikkoutumisen varalta. Opinnäytetyössä käytettävää lähdeaineistoa haettiin internetistä löytyvistä artikkeleista ja pelimoottorien valmistajien omilta sivuilta. Aineistoa säilytettiin Chrome-nettiselaimen kirjainmerkeissä. Tietokoneen C-asemalle luotiin kansio, jossa säilytettiin videoita ja kuvia demon luomisesta kummallakin pelimoottorilla. Kansiosta tehtiin varmuuskopiointi Teams-palvelimelle.