

Unreal Engine 5:n uudet ominaisuudet



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus

Syksy 2022

Ville Vainio

Opinnäytetyössä esitellään Epic Games-peliyhtiön pelimoottorin vuonna 2022 julkaistun version Unreal Engine 5:n uusia ominaisuuksia, sekä samalla tutustutaan moottorin käyttöön perustasolla. Opinnäytetyössä esitellään tärkeimpiä uusia ominaisuuksia ja niiden toimintaa, sekä miten niitä käytetään. Työssä myös selvitetään, miten moottorin editori on visuaalisesti ja toiminnallisesti muuttunut, ja miten nämä muutokset ovat parantaneet sen käyttöä.

Tutkimuskysymyksiin lähdettiin hakemaan vastauksia tutkimalla Unreal Engine 5:n Early Access-version dokumentaatiota, jossa esitellään uusia ominaisuuksia ja editorin muutoksia.

Opinnäytetyö on jaettu kahteen osaan, teoriaosuuteen ja käytännön osuuteen.

Teoriaosuudessa käydään läpi pelimoottorin tärkeimmät uudet ominaisuudet, selitetään niiden toimintaperiaatetta ja mietitään, mitä uusia mahdollisuuksia nämä ominaisuudet tuovat peleihin. Ominaisuuksien jälkeen selvitetään myös pelimoottorin editorin käyttöliittymän muutokset, sekä miten nämä edistävät pelien kehittämistä. Käytännön osuudessa lähdettiin luomaan pientä demoa, jossa tutustutaan editorin käyttöön pyrittiin hyödyntämään teoriaosassa esiteltyjä graafisia ja teknisiä ominaisuuksia.

Unreal Engine 5:n uudet graafiset menetelmät tuovat paljon apua pelikehittäjille paremman peligrafiikan tekoon. Naniten ja Lumenin avulla voidaan näyttää tarkkoja yksityiskohtia aidolla valaistuksella peleissä. Näistä voi olla hyötyä varsinkin virtuaalitodellisuudessa, joka on usein hyvin raskasta peruskäyttäjien tietokoneille. Moottorin uusi versio tuo myös uusia työkaluja äänen ja pelinsisäisen fysiikan parantamiseen, sekä helpottaa pelien kehittämistä muuttamalla editorin käyttöliittymää suoraviivaisemmaksi ja jokaisen käyttäjän omaan työskentelyyn mieleiseksi muokattavaksi. Opinnäytetyötä tehdessä, tekijä pääsi tutustumaan pelimoottorin käyttöön. Lukija voi myös oppia moottorin peruskäytöstä työtä lukemalla.

Degree Programme in Business Information Technology
Author Ville Vainio
Subject Unreal Engine 5's new features
Supervisors Tommi Lahti

Abstract
Year 2022

This thesis introduces the new features of Epic Games' game engine's new version, Unreal Engine 5, and at the same introduces the basic usage of the engine. The most important features and how they are used are demonstrated. The thesis also examined, how the user interface has changed, and how this affects game development.

The research started with examining Unreal Engine 5's Early Access documentation, which has information about new features and changes to the editor. The thesis is divided into two parts, which are the theory part and practical part. The most important new features and their mechanics are explained in the theory part, and what new possibilities they might offer for gaming. After discussing the features, the thesis investigates changes to the user interface, and how these might affect game development. In the practical part, a small demo is made, in which the basic usage of the engine is explored, and make use of the new graphic and technical features explained in the theory part.

Unreal Engine 5's most important graphic systems help game developers a lot in producing better game graphics. Nanite and Lumen, which are in the leading role in the documentation, make it possible to show extremely detailed images in games with realistic lighting. These two could prove to be very useful for virtual reality applications, which are usually quite resource heavy for basic users. The new version of the engine also brings new tools for making sounds and improving the in-game physics, and makes development easier with making the user interface more streamlined and customizable to the user's own needs. The thesis demonstrates the basics of using the game engine. Following this thesis, someone can make a basic game using Unreal Engine.

Keywords Unreal Engine, Game engine, Game Development

Pages 28 pages

Sanasto

Pelimoottori	Pelimoottorit ovat ohjelmistoja, joita käytetään pelien luomiseen.
Unreal Engine	Pelimoottori, mitä tässä opinnäytetyössä käytetään.
Early Access	Etujasssa julkaistu keskeneräinen versio ohjelmistosta.
C++	Ohjelmointikieli, mitä Unreal Engine käyttää.
Visuaalinen ohjelmointi	Ohjelmointia, jossa ei tarvitse kirjoittaa itse koodia, vaan voidaan yhdistellä valmiita funktioita yhteen.
Launcher	Käynnistysohjelma, joka toimii pelien ja ohjelmistojen jakelualustana.
Säteenseuranta	Valonsäteiden kulun simulaatio. Nykyaikaiset näytönohjaimet tukevat säteenseurantaa paremmin.
Skripti	Koodi, joka määrittää esimerkiksi pelihahmon toimintaa.
Polygon	3D-mallit muodostuvat pienistä kolmioista, joita kutsutaan polygoneiksi.
Level of detail	3D-mallien yksityiskohtien väheneminen, niiden ollessa kaukana.
Varjojen kartoitus	Pelimaailman varjojen säilytys tekstuurina.
Resoluutio	Kuvapisteen (pikselien) määrä näytöllä.
Asset	Digitaalinen asia, jolla on tietyt käyttöoikeudet. esimerkiksi: 3D-mallit, kuvat tai äänet
Collider	Peliesineen muoto, joka ei aina vastaa sen oikeaa muotoa. Käytetään fyysisissä tapahtumissa ja törmäyksissä

Sisälllys

1	Johdanto	1
2	Unreal Engine	2
2.1	Unreal Engine 5	2
2.1.1	Lisenssimaksut eli rojaltit	2
2.1.2	Alustat	2
3	Editori	3
3.1	Vaatimukset	3
3.2	Content Browser	4
3.3	Blueprint visuaalinen ohjelmointi	5
3.4	Actors	6
3.5	Pluginit eli liitännäiset	7
4	Unreal Engine 5:n lisäämät ominaisuudet	8
4.1	Nanite	8
4.2	Lumen	10
4.3	World Partition	11
4.4	MetaSounds	11
4.5	Virtual Shadow Maps	12
4.6	Temporal Super Resolution	12
4.7	Chaos Physics	13
5	Käyttöliittymän parannukset	14
6	Demon toteutus	16
6.1	Demon tavoitteet ja suunnitelma	16
6.2	Aloitus	16
6.3	Nanite	17
6.4	Lumen	20
6.5	Chaos Physics	21
6.6	World Partition	24
7	Demon lopputulos ja huomiot	27
7.1	Demon lopputulos	27
7.2	Editorin käyttö	27
7.3	Unreal Enginen dokumentaatio	27
8	Yhteenveto	28
	Lähteet	29

Kuvat ja taulukot

Kuva 1. Kuvakaappaus Content Drawerista.	5
Kuva 2. Blueprint-skripti. Painettaessa pelin sisäistä esinettä hiirellä, soitetaan ääni, luodaan partikkeleita ja tärisytetään näyttöä.	6
Kuva 3. Naniten Cluster Visualization. Ylhäällä malli ja alhaalla sen Clusterit	9
Kuva 4. Lumenin kortit visualisoituna komennolla r.Lumen.Visualize.CardPlacement 1.11	
Kuva 5. Unreal Engine 4:n editorin perusnäkyvä.	15
Kuva 6. Unreal Engine 5:n editorin perusnäkyvä.	15
Kuva 7. Valaistuksen asetukset. Keltaisella merkittynä valinnat.	17
Kuva 8. Korkean laadun malli vasemmalla ja Nanite-malli oikealla.	18
Kuva 9. Nanite Settings.....	18
Kuva 10. Nanite Visualization. Vasemmalla itse luotu Nanite. Oikealla Nanite-valmis malli. Mitä suurempi määrä clustereita, sitä tarkempi malli.	19
Kuva 11. Nanite-malleista rakennettu taso.....	19
Kuva 12. Lumenilla epäsuorasti valaistu luola.....	20
Kuva 13. Tason Lumen Kortit.....	21
Kuva 14. Fracture Editorissa uuden Fracturen luonti. Keltaisella merkitty painike.....	22
Kuva 15. Murtuvan esineen collider asetukset.	23
Kuva 16. Kuva, jossa yhdistettynä näkyy kuutio ennen ja jälkeen pelin simoloinnin.	24
Kuva 17. World Partition -projektiasetukset.....	25
Kuva 18. World Partition -paneeli	26
Taulukko 1. Unreal Enginen suositellut järjestelmävaatimukset (Epic Games, n.d.-d).	3
Taulukko 2. Työssä käytetty järjestelmäkoonpano	4

Liitteet

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

1 Johdanto

Tässä opinnäytetyössä tutustutaan Unreal Engine 5 -pelimoottorin uusiin ominaisuuksiin, käyttäen ohjelman early access eli ennakkojulkaisu -versiota. Ensimmäiseksi selvitetään, mikä Unreal Engine on ja mitä sillä voidaan tehdä. Ominaisuuksien lisäksi käydään myös läpi käyttöliittymämuutoksia ja vanhojen työkalujen parannuksia. Opinnäytetyössä tehdään demo, joka käyttää uusia ominaisuuksia, kuten Nanite, Lumen ja Chaos Physics.

Valmiita pelimoottoreita on julkisia ja yksityisiä. Yksityiset pelimoottorit ovat yhtiöiden omassa yksityiskäytössä. Julkiset pelimoottorit ovat kaikkien ladattavissa ja käytettävissä. Unreal Engine on yksi suosituimmista julkisista pelimoottoreista (Toftedahl & Engström, 2019, s. 8), ja sitä voi käyttää ilmaiseksi kuka tahansa. Valmiissa pelimoottoreissa on valmiiksi asennettuna moduuleja, kirjastoja ja työkaluja, mitä tarvitsee pelin tekemiseen, sekä laajat dokumentaatiot, kuinka niitä käytetään. Pelimoottorien kehitystyökaluja on usein helppo opetella ja usein internetistä löytyy ohjeita näiden käyttöön.

Opinnäytetyön tekijällä on jo kokemusta pelikehityksestä, mutta eri pelimoottoreilla. Tekijän motiivina toimii halu tutustua Unreal Enginen käyttöön, sekä oma kiinnostus pelikehitykseen

Opinnäytetyön lukeminen edellyttää peruskäsitystä ohjelmoinnista ja ohjelmistotyökalujen käytöstä. Unreal Enginessä ohjelmoidaan käyttämällä C++ tai käyttämällä visuaalista ohjelmointia, jossa kytketään valmiita tai itse tehtyjä funktioita yhteen.

Tämän opinnäytetyön tutkimuskysymyksiä ovat: Mitä uusia ominaisuuksia Unreal Engine 5 sisältää? Miten uudet ominaisuudet toimivat ja miten niitä käytetään? Miten käyttöliittymä on muuttunut ja miten tämä vaikuttaa editorin käyttöön?

2 Unreal Engine

Entiseltä nimeltään Epic MegaGames -peilyhtiö teki alun perin moottorin pelille Unreal vuonna 1998, josta se sai myös nimensä (Game Developer, 2009). Unreal Engine muuttui ilmaiseksi kolmannen version aikaan, ja kaikkien Unreal Enginen versioiden lähdekoodi on vapaata ja saatavilla GitHub-sivustolta, ja se on ohjelmoitu C++ -ohjelmointikielellä. (Epic Games, n.d.-a)

2.1 Unreal Engine 5

Unreal Engine 5 on uusin versio pelikehittäjä Epic Gamesin pelimoottorista, joka julkaistiin vuoden 2022 alkupuoliskolla. Tässä opinnäytetyössä on käytetty Unreal Engine 5:n Early Access -esiversiota, joka oli käytettävissä ennen virallista julkaisua. Early Access -versio sisältää tärkeimmät uudet ominaisuudet, pieniä muutoksia, kuten bugikorjauksia lukuunottamatta

2.1.1 Lisenssimaksut eli rojaltit

Pelien ja muiden projektien kehittäminen Unreal Enginellä on täysin ilmaista. Tuotteen julkaisu on myös ilmaista, kunnes se on tuottanut yli miljoona dollaria, jonka jälkeen Epic Games saa tuotoista 5 % rojalteina. Näin Epic Games perii lisenssimaksuja vain hyvin onnistuneista projekteista ilman haittaa pienemmille kehittäjille. (Epic Games, n.d.-b)

2.1.2 Alustat

Unreal Enginen editoria voidaan käyttää pelikehitykseen Windowsilla, macOS:llä ja Linuxilla. Pelejä voidaan julkaista seuraaville alustoille: Windows PC, Playstation 4, Playstation 5, Xbox One, Xbox Series X, Nintendo Switch, Google Stadia, macOS, iOS, Android, erilaiset AR alustat, useimmat VR alustat, Linux, SteamOS ja HTML5. (Epic Games, n.d.-a)

3 Editori

Unreal Enginen ja sen kehitystyökalut voi ladata Epic Games launcher -käynnistysohjelmasta, joka ladataan Epic Gamesin nettisivuilta. Uusin versio Epic Games launcherissa työn aloituksen aikaan oli Unreal Engine 4.27, mutta myös vanhempia versioita on ladattavissa, tosin on aina suositeltavaa uutta projektia aloittaessa käyttää uusinta vakaata versiota. Versiota ei myöskään kannata vaihtaa kesken projektin ilman tärkeää syytä mahdollisten ongelmien takia. Käynnistysohjelmasta voi myös ladata Unreal Engine 5:n Early Access -version, johon tullaan tässä opinnäytetyössä keskittymään.

3.1 Vaatimukset

Unreal Enginen editorin käyttäminen vaatii Epic Games tilin ja minimi-järjestelmävaatimusten täyttymisen (Taulukko 1). Tilin voi luoda yhtiön nettisivuilla tai Epic Games Launcherissa. Järjestelmävaatimukset ohjelmiston käyttöön ovat samat, kuin aiemmassa versiossa. On yleensä suositettavaa täyttää suositellut vaatimukset, eikä minimivaatimuksia, muuten työnteko voi kärsiä. Työssä käytetty tietokone täyttää vaatimukset ja mahdollistaa myös säteenseurannan käytön projektissa (Taulukko 2). Unreal Engine 5:een on ladattavissa myös demoprojekti Valley of the Ancients, joka luotiin moottorin uusien ominaisuuksien stressitestaamiseen, joten sen avulla voi myös helposti tutustua uutuuksiin. Valley of the Ancientsin järjestelmävaatimukset ovat erittäin korkeat raskaan grafiikan vuoksi, ja ne ovat listattu Unreal Enginen sivuilla. (Epic Games, n.d.-c)

Taulukko 1. Unreal Enginen suositellut järjestelmävaatimukset (Epic Games, n.d.-d).

Käyttöjärjestelmä	Windows 10 64-bit
Proessori	Neliytiminen Intel tai AMD, 2.5 GHz tai enemmän
Muisti	8GB muistia

Näytönohjain	DirectX 11 tai DirectX 12 yhteensopiva näytönohjain
---------------------	---

Taulukko 2. Työssä käytetty järjestelmäkokoontu

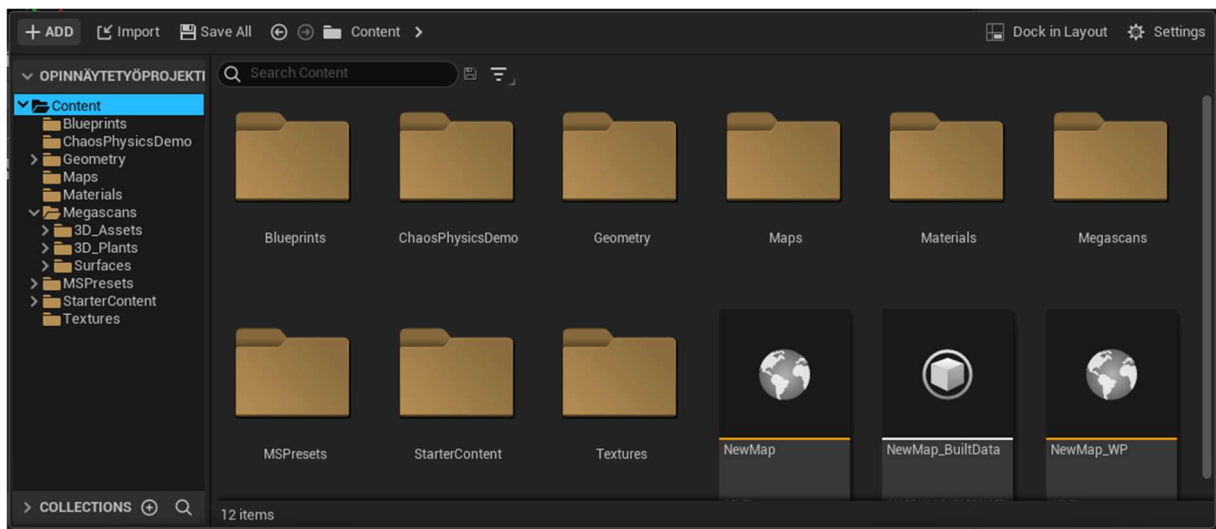
Käyttöjärjestelmä	Windows 10 Home
Proessori	AMD Ryzen 7 3700X 4.4 GHz
Muisti	16 GB
Näytönohjain	Asus Nvidia GeForce RTX 3070

3.2 Content Browser

Content browserilla, eli sisällön selaimella voidaan hallita kaikkea projektin sisältöä. Selaimen voidaan luoda, tuoda, järjestää, katsoa tai muokata projektin tiedostoja. Selaimella voidaan luoda kansioita, joihin pelin eri osa-alueiden tiedostot voidaan sijoittaa. Hakupalkilla voidaan hakea tiedostoja niiden nimellä. Selain osoittaa, jos jossain tiedostoissa on ongelmia. Content browserin voi avata yläpalkin Window-luettelosta. Selaimia voi olla auki yhtäaikaaisesti neljä, jos käyttäjän tarvitsee nähdä monta kansiota yhtäaikaisesti tai siirrellä tiedostoja niiden välillä.

Unreal Engine 5:ssä on Content Drawer, joka toimii Content Browserin erillisenä instanssina. (Kuva 1) Content Drawerin voi avata pikanäppäinyhdistelmällä Ctrl+Välilyönti, jonka jälkeen se aukeaa editorin alareunaan ja sulkeutuu automaattisesti käytön jälkeen tai klikattaessa muualle editorissa. Drawerin saa myös pysymään koko ajan auki oikean yläkulman dock-napista. Käyttämällä Content Draweria, voidaan maksimoida editorin pelinäkömön koko piilottamalla sisältöselain, kun sitä ei tarvita.

Kuva 1. Kuvakaappaus Content Drawerista.

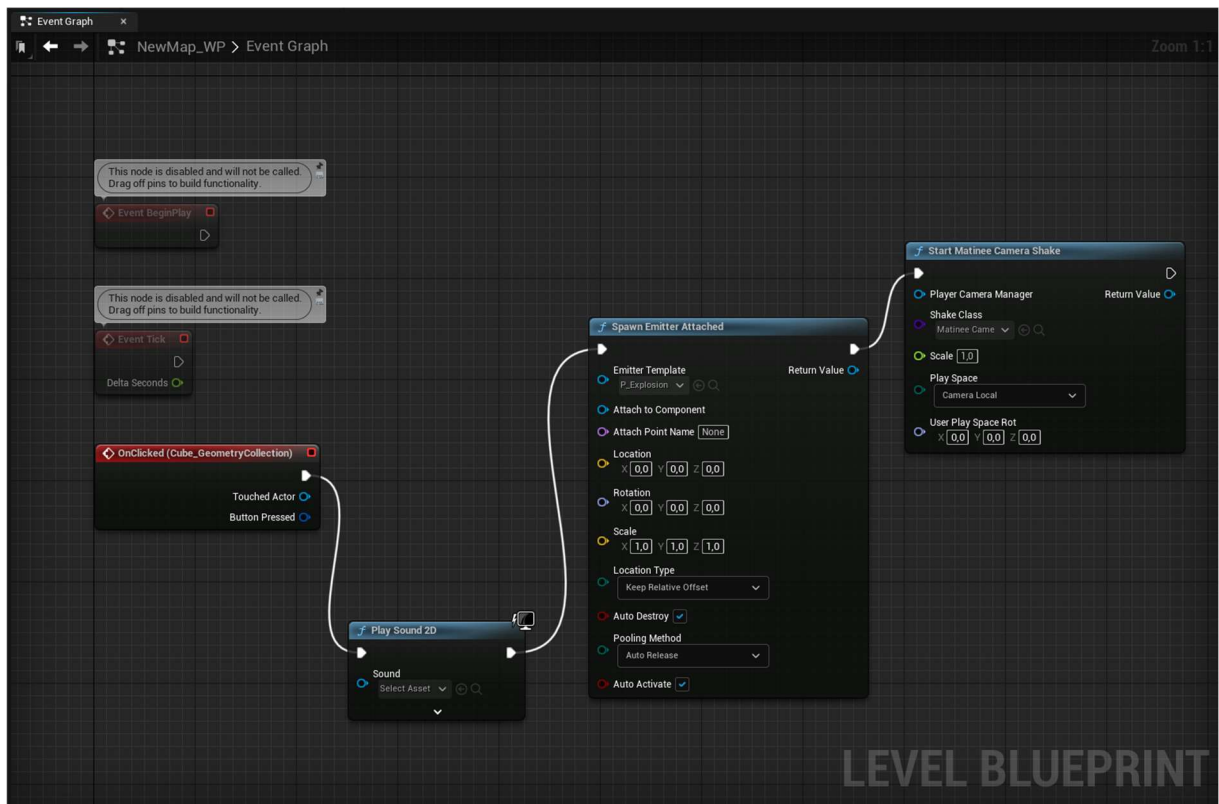


3.3 Blueprint visuaalinen ohjelmointi

Blueprintit ovat Unreal Engineissä oleva skriptejä. Blueprunteilla voidaan ohjelmoida pelilogiikkaa täysin visuaalisesti ja niillä voi kokemattomampi henkilö luoda toimintoja, joita normaalisti vain ohjelmoija osaisi tehdä. Järjestelmässä on noodipohjainen käyttöliittymä, jossa skriptit toimivat, kuin rakennuspalikat. (Kuva 2) Näitä rakennuspalikoita yhdistelemällä voidaan luoda monenlaisia toimintoja peliin. Unreal Engine sisältää valmiita skriptejä esimerkiksi hahmon liikkeisiin ja perustoimintoihin kuten peliobjektien yhdessä toimintaan.

Blueprinteista voidaan luoda luokkia, jotka voivat tämän jälkeen lisätä toimintoja valmiiden peliluokkien päälle tai toimia objekteina pelin sisällä. Blueprint-luokat ovat hyviä interaktiivisten toimintojen luontiin, kuten oven avaamiseen napilla, keräiltävien esineiden hallintaan tai tapahtumaketjuihin. (Epic Games, n.d.-e)

Kuva 2. Blueprint-skripti. Painettaessa pelin sisäistä esinettä hiirellä, soitetaan ääni, luodaan partikkeleita ja tärisytetään näyttöä.



3.4 Actors

Kaikki esineet, joita voidaan asettaa maailmaan ovat actoreita. Näitä ovat esimerkiksi kamerat, staattiset polygoniverkot tai pelaajien aloituspaikat. Actoreita voidaan muokata erilaisilla muuttujilla komponenttien avulla, kuten pyörittäminen tai skaala. Actoreita voidaan luoda ja tuhota pelin koodin kautta. Kun actor on tuhottu, se siirtyy odottamaan pelimoottorin seuraavaa roskankeruuta, jolloin se poistuu lopullisesti. Jokainen actor päivittyy oletuksena jokaisella kuvaruudun päivittymisellä, mutta tätä aikaväliä voidaan muuttaa erilaiseksi. Moninpeleissä actoreita ja niiden arvoja voidaan replikoida muille pelaajille, jotta kaikkien peli pysyisi samassa tahdissa.

Actorit muodostuvat komponenteista, jotka voivat määrätä miten ne liikkuvat tai piirtyvät näytölle. Komponentteina voi olla esimerkiksi sellaisia, jotka sisältävät actorin toimivuuteen liittyvää koodia, niiden sijaintiin ja asentoon liittyviä tietoja tai siihen graafisen mallin liittäviä. (Epic Games, n.d.-f)

3.5 Pluginit eli liitännäiset

Useimmissa pelimoottoreissa voidaan asentaa liitännäisiä, jotka lisäävät editoriin tai peliin toimintoja ja ominaisuuksia, muuttamatta moottorin koodia. Liitännäisillä voidaan saada esimerkiksi lisää valikoita editoriin, erilaisia funktioita pelin toimintaan tai jopa täysin uusia ominaisuuksia, kuten animaatiojärjestelmiä.

Liitännäiset ovat projektikohtaisia ja niitä voidaan lisätä projektiin Unreal Engine Marketplacen kautta. Monet Unreal Enginen työkalut ovat valmiiksi tehty helposti laajennettaviksi liitännäisillä. Kuka tahansa käyttäjä voi tehdä, sekä julkaista ja myydä liitännäisiä Unreal Marketplacessa.

4 Unreal Engine 5:n lisäämät ominaisuudet

Unreal Engine 5 pyrkii nostamaan pelikehityksen seuraavalle sukupolvelle sen uusilla ominaisuuksilla. Uudet graafiset ominaisuudet, kuten Nanite ja Lumen mahdollistavat realistisemmän peligrafiikan näyttämistä reaaliajassa paremmalla suorituskyvyllä.

Avoimet maailmat ovat olleet ennen kooltaan rajatumpia. Unreal Engine 5 sisältää uusia työkaluja pelimaailmojen jakamiseen osiin, joka parantaa suorituskykyä, sekä helpottaa pelikehittäjien työskentelyä keskenään mahdollistaen työskentelyn eri alueille samanaikaisesti.

Uudet käyttöliittymän muutokset ja visuaalisen työskentelyn integrointi moottorin sisälle nopeuttavat työskentelyä, sekä vähentää tarvetta ulkopuolisten ohjelmien käyttöön samanaikaisesti, sekä sisältää uusia työkaluja konsolien ja mobiililaitteiden kanssa työskentelyyn. (Epic Games, 2021)

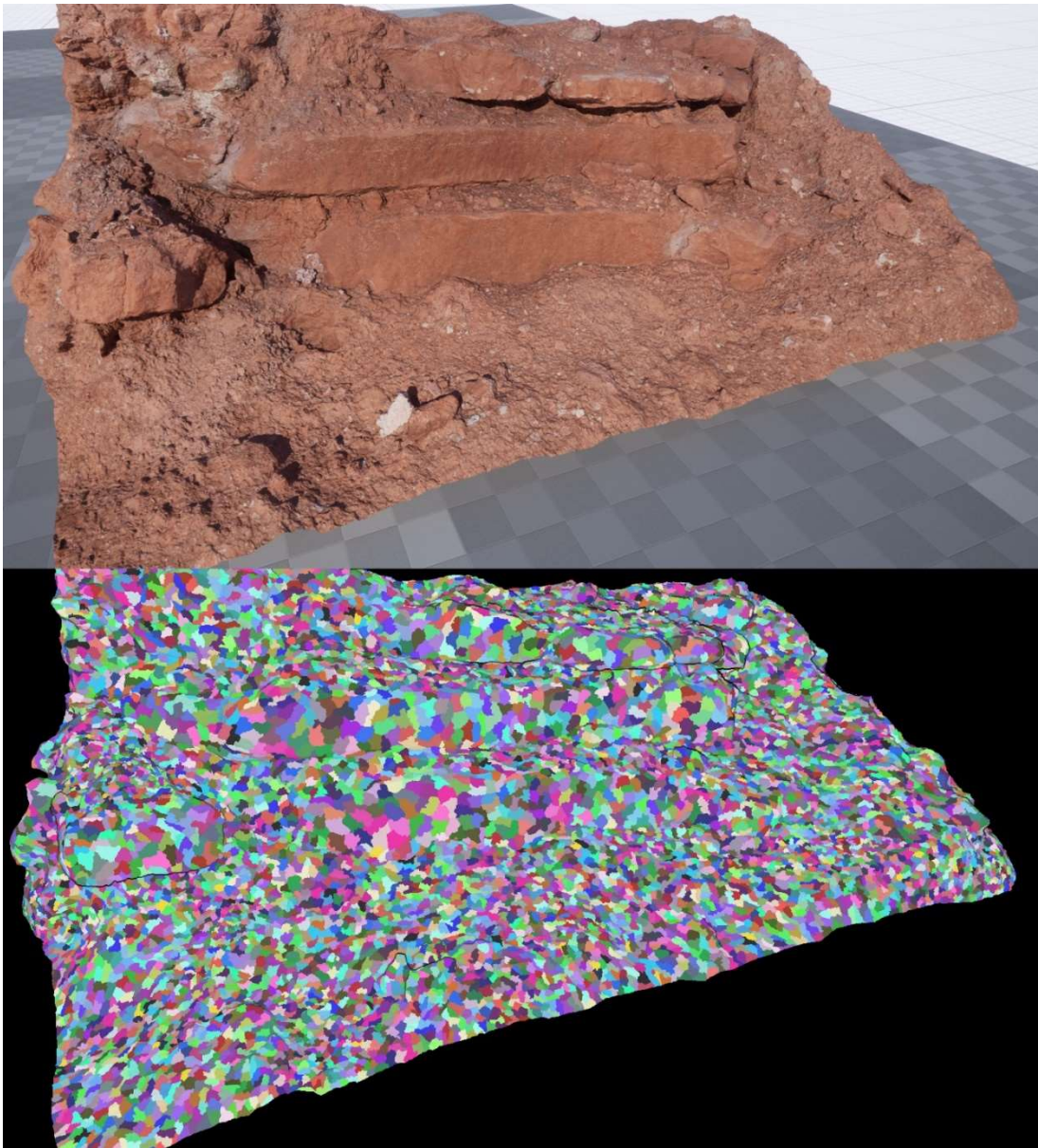
Tässä luvussa esitellään dokumentaatiossa esitellyimpiä merkittävimpiä ominaisuuksia, jotka ovat Nanite, Lumen, World Partition, MetaSounds, Virtual Shadow Maps, Temporal Super Resolution ja Chaos Physics, ja kerrotaan niiden toiminnasta. Ominaisuuksien tutkimisen yhteydessä on tehty myös ohessa kokeiluja editorin sisällä ja otettu esimerkkikuvia tekstin avuksi.

4.1 Nanite

Nanite Virtualized Geometry on uusi 3D-mallien piirtomenetelmä, joka mahdollistaa hyvin tarkkojen yksityiskohtien piirtämisen reaaliajassa hyvällä suorituskyvyllä, sekä useampia 3D-malleja ruudulla samanaikaisesti. Menetelmällä 3D-malli jaetaan osiin eli clustereihin, jolloin voidaan muuttaa esineiden tarkkuutta clustereittain. Vain mallien clusterit, jotka ovat näkyvissä ja tarpeeksi lähellä piirretään tarkemmin. (kuva 3) Käyttäen Nanitea, voidaan käyttää tarkimpia 3D-malleja, jotka voivat sisältää jopa miljoonia polygoneja. Level of Detail mallit ovat cluster-kohtaisia ja automaattisesti käsiteltyjä, eikä niitä tarvitse enää manuaalisesti tehdä.

Naniteen voi käyttää mitä tahansa malleja, ja sen käyttäminen vaatii vain yhden rastin ruutuun ja moottori tekee lopun työn. 3D-mallit jaetaan osioihin ja tarkkuuden taso on osiokohtaista, ja Level of Detail mallit vaihtuvat lennossa saumattomasti tarpeen mukaan pyrkien näytön pikselien tarkkuuteen. Jos siirrytään aivan kiinni johonkin mallin osaan, tämä voidaan piirtää täysin alkuperäisellä tarkkuudella säilyttäen suorituskyky huonontamalla malleja, jotka eivät ole sillä hetkellä tärkeitä. Nanitesta voi olla erityisesti hyötyä virtuaalitodellisuudessa, jossa suorituskyky on usein ongelma ja esineitä pystytään katsomaan hyvin läheltä, jolloin huonommat mallit pelin sisällä tulevat selkeämmiksi. (Epic Games, n.d.-g)

Kuva 3. Naniten Cluster Visualization. Ylhäällä malli ja alhaalla sen Clusterit



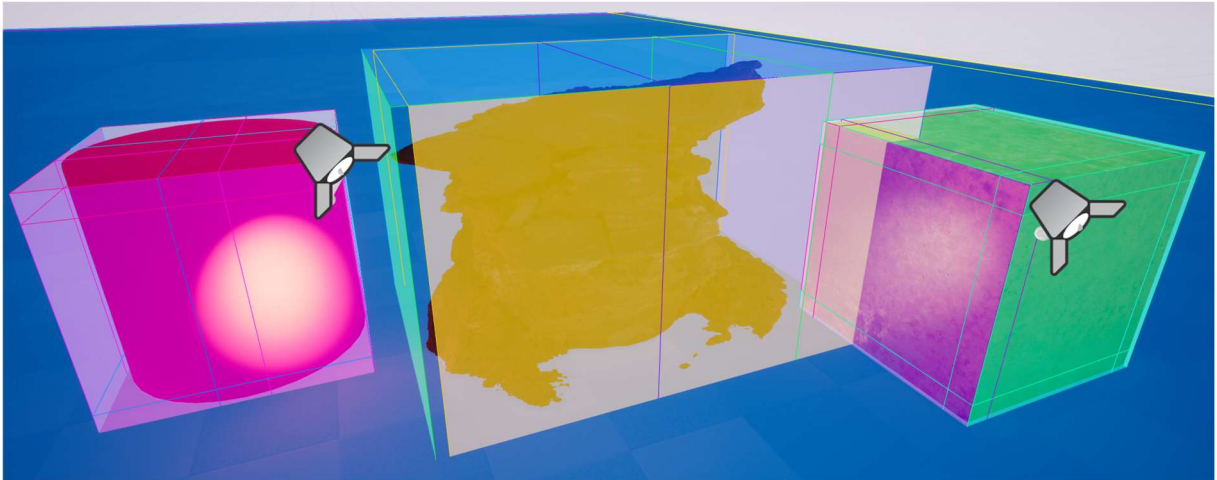
4.2 Lumen

Lumen Global Illumination and Reflections on Unreal Engine 5:n uusi valaistus järjestelmä, joka on suunnattu seuraavan sukupolven konsoleille ja tietokoneille. Ominaisuus on myös lisätty aikaisemmin Unreal Engine 4:ään beta-versiona, mutta se pitää aktivoida projekteissa erikseen ja saattaa sisältää paljon bugeja. Unreal Engine 4:ssä oleva Screen Space Global Illumination, joka toimi heijastaen ainoastaan näkökentässä olevat asiat oli epäluotettava, ja uudet säteenseurantamenetelmät voivat olla liian raskaita. Lumen pyrkii korjaamaan nämä ongelmakohdat. (Epic Games, n.d.-h)

Global Illumination eli globaali valaistus heijastaa valoa pinnoista muihin pintoihin. Heijastukset ottavat mukaan myös pintojen värit ja tätä efektiä kutsutaan värivuodoksi. Esineet maailmassa myös estävät epäsuoraa valoa ja tuottavat täten omia varjoja. Lumen tarjoaa rajattoman määrän valonsäteiden kimmutuksia, joita tarvitaan varsinkin kirkkaiden värien kanssa. Lumenin toimiessa Naniten kanssa yhdessä saadaan tarkkoja normaalikartoituksia. Järjestelmä tukee kaikkia heijastavia pintoja riippumatta ottaen huomioon kaikki asetukset, kuten epätasaisuudet, sekä erittäin kiiltäviä lakkapintoja. Kimmotettu epäsuora valo luo myös omat heijastuksensa.

Lumen käyttää ohjelmistotason säteenseurantaa millä tahansa laitteella, mutta voi saavuttaa vielä parempia tuloksia säteenseurantaa tukevilla näytönohjaimilla, jos laitteistotason säteenseuranta laitetaan päälle. Järjestelmä luo välimuistiin niin sanottuja kortteja pinnoista, joita käytetään nopeasti valaistuksen havaitsemiseen valonsäteiden osumispisteissä. Lumen katsoo jokaisen esineen materiaalien ominaisuudet useasta kulmasta ja tallentaa ne tähän sijaintiin, joka voidaan visualisoida editorissa (kuva 4). Kun nämä kortit on laskelmoitu, voidaan nopeasti laskea suora ja epäsuora valo näissä sijainneissa. (Epic Games, n.d.-i)

Kuva 4. Lumenin kortit visualisoituna komennolla r.Lumen.Visualize.CardPlacement 1.



4.3 World Partition

World Partition on uusi järjestelmä, joka jakaa isot pelimaailmat ruudukkosoluihin, joita ladataan vain tarvittaessa, säästäten resursseja. Ennen maailmat piti jakaa alitasoihin, joita ladattiin maailmassa kulkiessa. Tämä aiheutti ongelmia tiedostojen jaossa käyttäjien välillä ja koko maailman katsominen samaan aikaan oli haasteellista. Uusi World Partition - järjestelmä pitää koko maailman samassa tiedostossa. Maailman solut voidaan ladata pelattaessa automaattisella järjestelmällä riippuen etäisyydestä pelaajaan tai muuhun streaming sourceksi asetettuun esineeseen, eli esine, jonka läsnäolo lataa solun. Actorit ovat myös säilytetty omassa tiedostossaan tason sijaan, jolloin muut työntekijät tiimissä voivat muokata actoreita ja tasoa erikseen. World Partition myös tarjoaa editorin sisäisen kartan kameran sijainnin näyttämiseen, helpottaen kehittäjien työskentelyä. (Epic Games, n.d.-j)

4.4 MetaSounds

MetaSounds on Unreal Engine 5:n uusi menetelmä äänien hallinnalle, joka tulee korvaamaan aiemman Sound Cues järjestelmän. Vanhassa järjestelmässä oli rajalliset ominaisuudet ja uusi menetelmä mahdollistaa täyden digitaalisen signaalinkäsittelyn äänilähteille. MetaSounds pystyy generoimaan synteettistä ääntä ja niitä voidaan sekoittaa vapaasti muiden äänien kanssa. Järjestelmä voidaan myös integroida helpommin pelidatan ja

pelaajan vuorovaikutusten kanssa, esimerkiksi äänien muuttuminen pelin tapahtumien mukaan.

MetaSoundsilla on täysin visuaalinen noodipohjainen käyttöliittymä, jota äänisuunnittelijat voivat käyttää ilman ohjelmointikokemusta. Jokaista ääntä voidaan hienosäätää näytteenottotaajuuden ja puskurikoon, jotta voidaan saavuttaa haluttu latenssi ja suorituskyky. (Epic Games, n.d.-k)

4.5 Virtual Shadow Maps

Virtual Shadow Maps on uusi varjojen kartoitusmetodi, jolla saadaan tarkempia korkean resoluution varjoja, jotka toimivat tarkempien mallien ja suurempien maailmojen kanssa. Varjot ovat tehty toimimaan yhteistyössä Naniten, Lumenin ja World Partitionin kanssa. Järjestelmä pyrkii nostamaan varjojen resoluutiota toimiakseen Naniten elokuvatason 3D-mallien kanssa, tarjoamaan pehmeitä varjoja, tekemään yksinkertaisen ratkaisun varjoihin ilman suurempaa konfigurointia ja yhdistämään useita paikallaan olevien valojen ratkaisuja yhdeksi.

Shadow Map Ray Tracing näytteenottoalgoritmi on kevyempi, kuin normaalit säteenseurannat ja tuottaa luotettavempia pehmeitä varjoja esineiden ollessa kauempana kohteesta, johon varjo muodostuu. Valonlähteiden kokoa voidaan myös muuttaa, joka tekee varjojen reunoista eli penumbroista pehmeämpiä. (Epic Games, n.d.-l)

4.6 Temporal Super Resolution

Temporal Super Resolution on uusi Epic Gamesin luoma kuvan reunanpehmennysalgoritmi, joka pyrkii korvaamaan aiemman TemporalAA reunanpehmennyksen.

Reunanpehmennysmenetelmät pyrkivät vähentämään sahalaitaisia esineiden reunoja, joita esiintyy näyttöjen vähäisten pikselimäärien takia. Sahalaitaisuus ei ole ongelma suuremmilla resoluutioilla, mutta tämä vaatii moninkertaisesti enemmän suorituskykyä. Uusi menetelmä toimii piirtämällä kuva alemmalla resoluutiolla, joka skaalataan tarkemmalle resoluutiolle, jolloin saadaan korkeampi tarkkuus paremmalla suorituskyvyllä. Tämän tyylisiä Temporal

Anti-Aliasing eli TAA algoritmeja on jo paljon, ja tämä on Epic Gamesin uusi oma versio. (Epic Games, 2021)

4.7 Chaos Physics

Chaos Physics on uusi kevyt fysiikkasimulaatiomenetelmä Unreal Engine 5:ssä, joka on rakennettu seuraavan sukupolven pelit mielessä. Se on aikaisemmin ollut myös käytettävissä Unreal Engine 4:ssä betaversiona, ja sitä on käytetty muun muassa suosituissa pelissä, Fortnitessä. Unreal Engine 5:n julkaisu tuo mukanaan täyden valmiin version. Tuettuja ominaisuuksia ovat mm. esinefysiikka, esineiden animaatiot, kangasfysiikat, esineiden tuhoutuminen, räsynukkefysiikat, ajoneuvot, fysiikkakentät, nestefysiikka ja pelihahmojen hiusten simulaatio. Chaos Physics korvaa näillä vanhemman PhysX-fysiikkamoottorin kyseiset ominaisuudet ja parantaa näiden tarkkuutta ja suorituskykyä. Fyysisille objekteille on saatavilla vielä kokeiluvaiheessa oleva epäsynkroninen tahdistus, eli simulaatio laskelmoidaan erillään pelin muista järjestelmistä, parantaen luotettavuutta ilman haittaa muulle pelille.

Kangasfysiikka on paranneltu ja sen konfiguroinnissa on enemmän ja vähemmän rajattuja muuttujia, joilla kehittäjä voi paremmin tähdätä laatuun tai suorituskykyyn. Tuulen simulointi ottaa nyt huomioon kappaleen pinta-alan. Kangasfysiikan toteutusta on myös helpotettu uusilla työkaluilla.

Chaos Destruction järjestelmä on kokoelma eri työkaluja realistiseen peliesineiden tuhoamiseen pienemmiksi osiksi reaaliajassa, sekä paremmalla suorituskyvyllä. Järjestelmä käyttää niin sanottuja Geometry Collectionseja, jotka ovat useista pienistä palasista rakennettuja kokoelmia, muodostaen kokonaisia esineitä, kunnes ne hajotetaan. Näiden kokoelmien halkaisemista ja hajottamista voidaan hallita tasoissa, jotta voidaan tuhota esineitä asteittain eri osissa. Fysiikan näyttävyyttä ja suorituskykyä voidaan parantaa tai huonontaa esineiden asetuksista hallitsemalla simulaation tarkkuutta, sekä antaa esineille eri vahinkorajoja. Chaos Destructionissa on myös välimuistijärjestelmä, mahdollistaen fysiikkatapahtumien tallentamista muistiin ja uudelleen näyttämistä ilman reaaliaikaista simulaatiota, jolloin pelin suorituskyky ei kärsi. (Epic Games, n.d.-m)

5 Käyttöliittymän parannukset

Epic Games pyrkii parantamaan editorin käyttöliittymää modernimpaan suuntaan ja parantamaan käyttäjien työskentelyä. Unreal Engine 5 editorin esiversiossa on jo tähän mennessä saatu useita muutoksia ja lisää on luvattu tulevaisuudessa, jopa virallisen julkaisun jälkeen. Ilmeisin muutos on perusnäkyvä, jossa on keskitytty piilottamaan turhat harhautukset ja maksimoimaan viewportin eli tasoeditorin näkymän koko. Kuvakkeet ovat suunniteltu uudelleen ja niistä on tehty pienempiä, sekä valikot on järjestetty paremmin. Paneeleista voi nyt luoda välilehtiä sivupalkeille. Kuvassa 5 on Unreal Engine 4:n perusnäkyvä, jota verrattaessa Kuvan 6, jossa on Unreal Engine 5:n perusnäkyvä, voidaan huomata hyvin suuri muutos näiden välillä. Käyttöliittymä on tummempi ja napit ovat erilaisia ja eri paikoissa. Unreal Engine 5 näyttää modernimmalta kuin Unreal Engine 4, joka vaikuttaa paljon vanhemmalta ja graafisesti yksinkertaiselta.

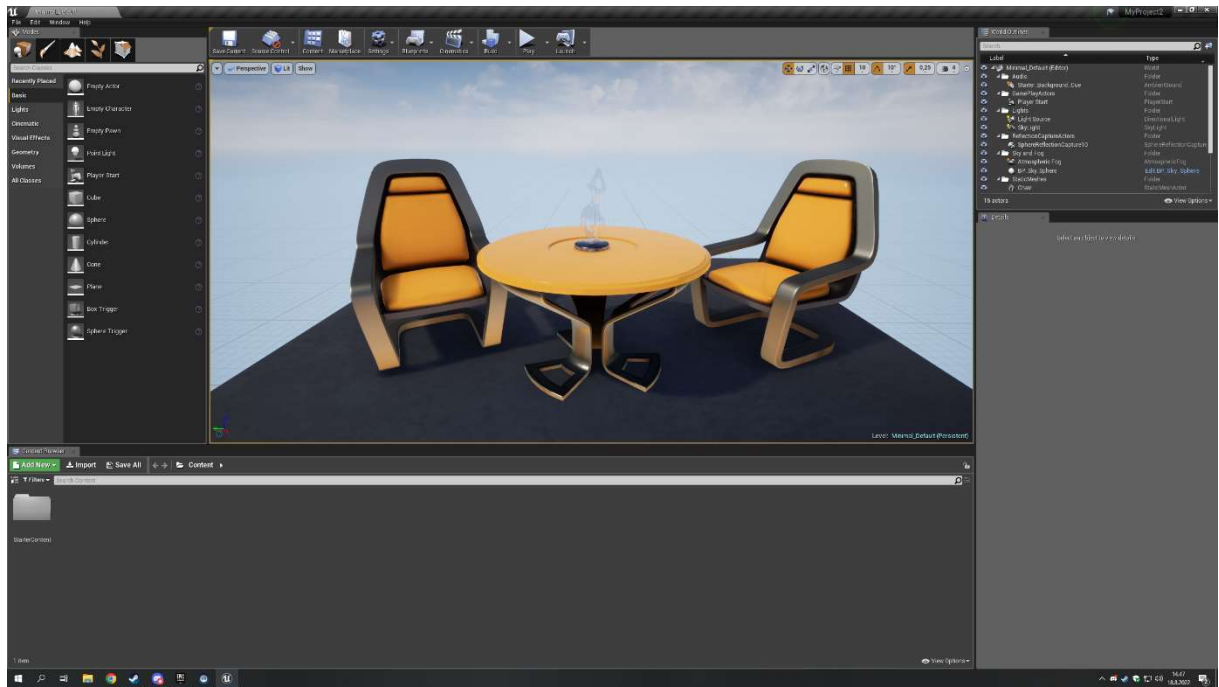
Editorin yläpalkkia on muokattu siten, että on kaksi luettelopainiketta Create ja Content. Create painikkeesta avataan luettelo, josta lisätään actoreita projektiin, ja tämän create-menun voi myös avata paneelina näkymään koko ajan. Content nappi yhdistää Unreal Engine 4:stä Content ja Marketplace napit yhdeksi ja sieltä voidaan avata sisältöselaimet tai hakea lisää sisältöä Marketplacesta tai Quixel Bridgestä. Alapalkissa ennen ollut modes-luettelo on nyt korvattu erillisillä napeilla jokaiselle modelle, jotka muuttavat viewportin eri tiloja.

Minkä tahansa paneelin voi siirtää sivupalkkiin välilehdeksi oikeaklikkaamalla sitä ja valitsemalla move to sidebar. Sivupalkissa oleva paneeli aukeaa väliaikaisesti painamalla, ja se sulkeutuu painaessa sen ulkopuolelle. Painamalla Ctrl ja välilyöntiä, aukeaa uusi Content Drawer, josta voidaan raahata esineitä peliympäristöön. Content Drawer sulkeutuu esinettä raahatessa. Tasoeditorissa voidaan lisätä ominaisuuksia suosikkeihin. Kun ominaisuuksia on suosikeissa, ne näkyvät aina paneelin yläosassa omana välilehtenään.

Unreal Engine 5:n editoria voi myös muokata visuaalisesti vaihtamalla eri elementtien värejä, joista voidaan luoda teemoja. Teemoja voi luoda omia tai muokata jo valmiina olevia. Editorin latausruutuun on lisätty vasempaan alakulmaan viesti, josta näkee mitä ladataan, sekä tasoeditorissa on latauspalkit eri prosesseille. Projektien luomisesta on tehty helpompaa ja kaikki tarvittavat asetukset on tiivistetty yhdelle sivulle.

Kun editori on paljon muokattavissa käyttäjän omien mieltymysten mukaiseksi, parantaa se jokaisen yksilön työskentelyä ja tuottavuutta. Kun saadaan näkymään monta eri työkalua samanaikaisesti eri käyttötarkoituksiin, saa tiimin jokainen eri työtehtävässä oleva työntekijä sopivan ympäristön työlleen. (Epic Games, n.d.-n)

Kuva 5. Unreal Engine 4:n editorin perusnäkö.



Kuva 6. Unreal Engine 5:n editorin perusnäkö.



6 Demon toteutus

6.1 Demon tavoitteet ja suunnitelma

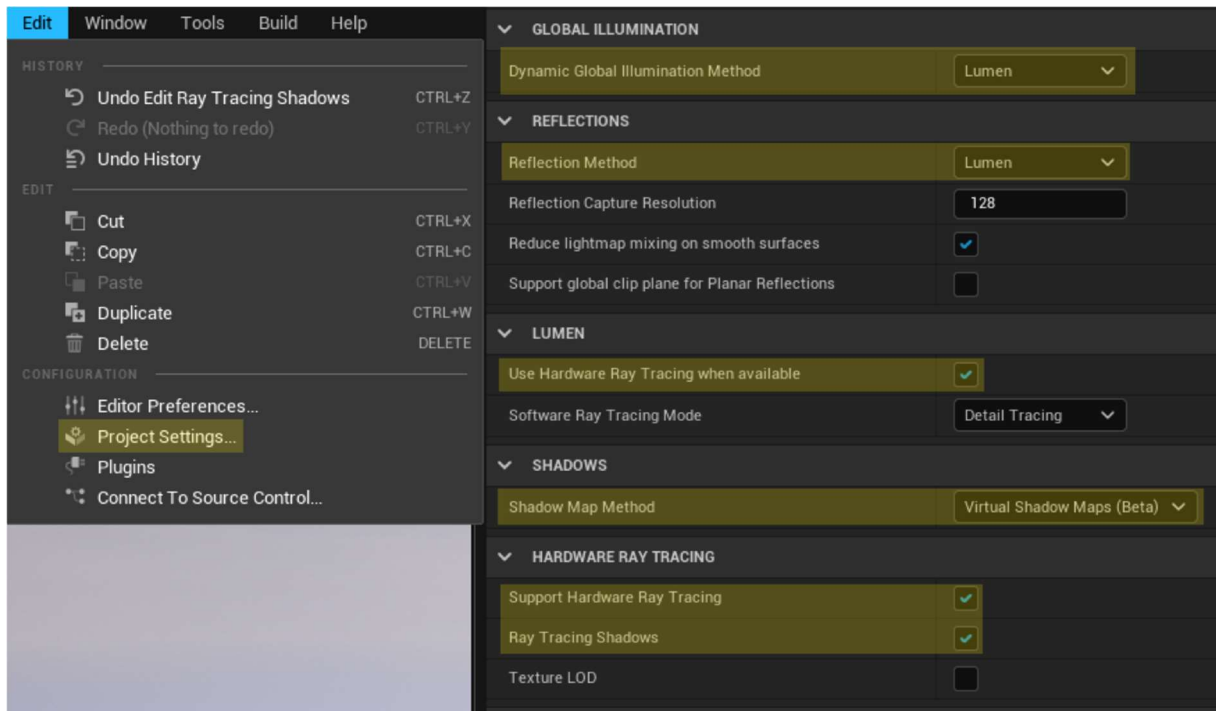
Tässä opinnäytetyössä on tavoitteena käytännön osuudessa luoda pieni demo, jossa käytetään teoriaosassa mainittuja uusia ominaisuuksia. Demossa tulee olemaan yksinkertainen taso, jossa käytetään assetteja Unreal Marketplacesta. Samalla harjoitellaan Unreal Editorin käyttöä. Opinnäytetyön tekijä on lukenut Unreal Enginen dokumentaatioita, sekä katsonut muutamia videoita editorin käytöstä ennen projektin aloittamista. Projektin tekoon käytettävässä järjestelmässä on säteenseurantaa tukeva näytönohjain, joten projektissa tullaan käyttämään säteenseurantaa.

6.2 Aloitus

Unreal Engine 5 Early Access versio on ladattu ja se käynnistetään Epic Games Launcherin Unreal Engine -välilehdeltä. Unreal Project Browserista valitaan tyhjä alusta projektille. Annetaan projektille jokin nimi ja projektin asetuksista valitaan Blueprint, maximum quality preset, starter content ja raytracing. Kaikki uudet ominaisuudet ovat projekteissa jo valmiina Unreal Engine 5:ssä, eikä tarvitse asentaa erikseen plugineina. Osa ominaisuuksista pitää laittaa erikseen päälle projektin asetuksista.

Demoa varten pitää hakea assetteja Marketplacesta. Tähän projektiin valittiin Quixel Megascan 3D-malleja niiden suuren polygon-määrän ja tarkkojen yksityiskohtien vuoksi. Lumen ja virtual shadow maps ovat valmiiksi päällä, kuten asetuksista näkyy. Hardware Ray Tracing alavalikosta täytyy laittaa erikseen Ray Tracing Shadows päälle, sekä Lumenin Hardware Ray Tracing. Näitä asetuksia vaihtaessa ensimmäistä kertaa vaaditaan Unreal Editorin uudelleenkäynnistys. Käynnistyessään uudestaan editori luo shaderit ja kortit Lumenille. Tässä voi kestää kauan. (Kuva 7)

Kuva 7. Valaistuksen asetukset. Keltaisella merkittynä valinnat.

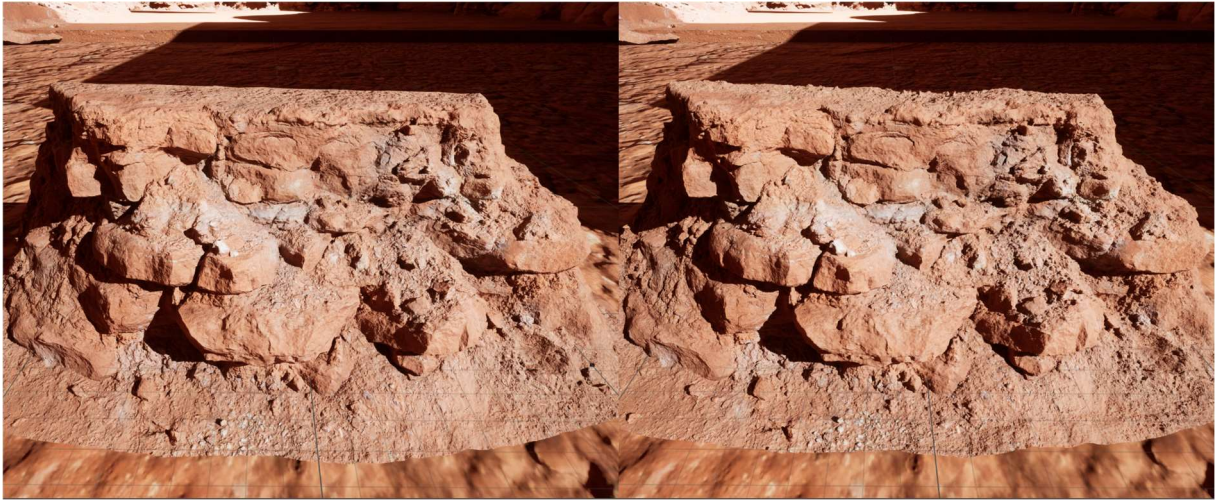


6.3 Nanite

Valley of the Ancients demossa käytetään Quixel Megascansin Canyons of Utah pakettia, jota tullaan tässä projektissa myös käyttämään. Quixel Megascans luonnosta skannatut mallit ovat kaikille käytettävissä olevia ilmaisia malleja. Quixelin Megascans -malleja voidaan ladata editorin sisältä Quixel Bridgen kautta, jonka voi avata editorin yläpalkin Content-valikosta. Quixel bridgessä valitaan mallien tarkkuus latausvalikosta, sekä pystyy suoraan lataamaan Nanite-valmiita malleja. Korkeimman laadun malli on silti huomattavasti epätarkempi, kuin Nanite-malli.

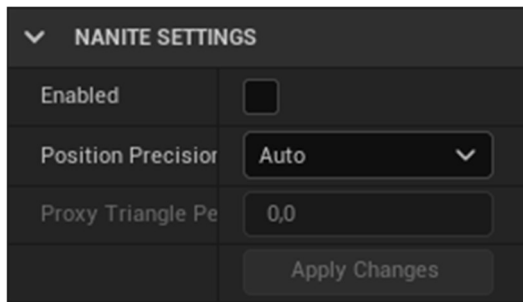
Projektiin ladattiin ja lisättiin samasta 3D-mallista korkean laadun versio ja Nanite-versio. Nanite näytti heti alussa tarkemmalta ja mallissa oli huomattavasti enemmän rosoisuutta ja syvyyttä. (Kuva 8)

Kuva 8. Korkean laadun malli vasemmalla ja Nanite-malli oikealla.



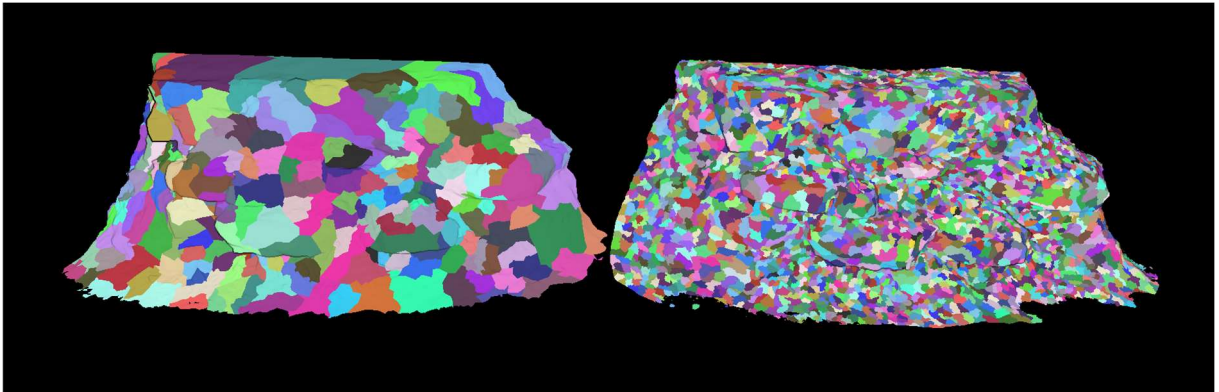
Muutetaan myös toinen malli käyttämään Nanitea. Valitaan actor, oikea klikataan sitä ja valitaan edit. Auenneessa static mesh editorissa on oikealla puolella asetuksia, joista toiseksi ylämpänä on Nanite asetukset (Kuva 9). Vaihdetaan Nanite päälle ja painetaan Apply Changes, jolloin editori tekee automaattisesti mallista Nanite meshin.

Kuva 9. Nanite Settings



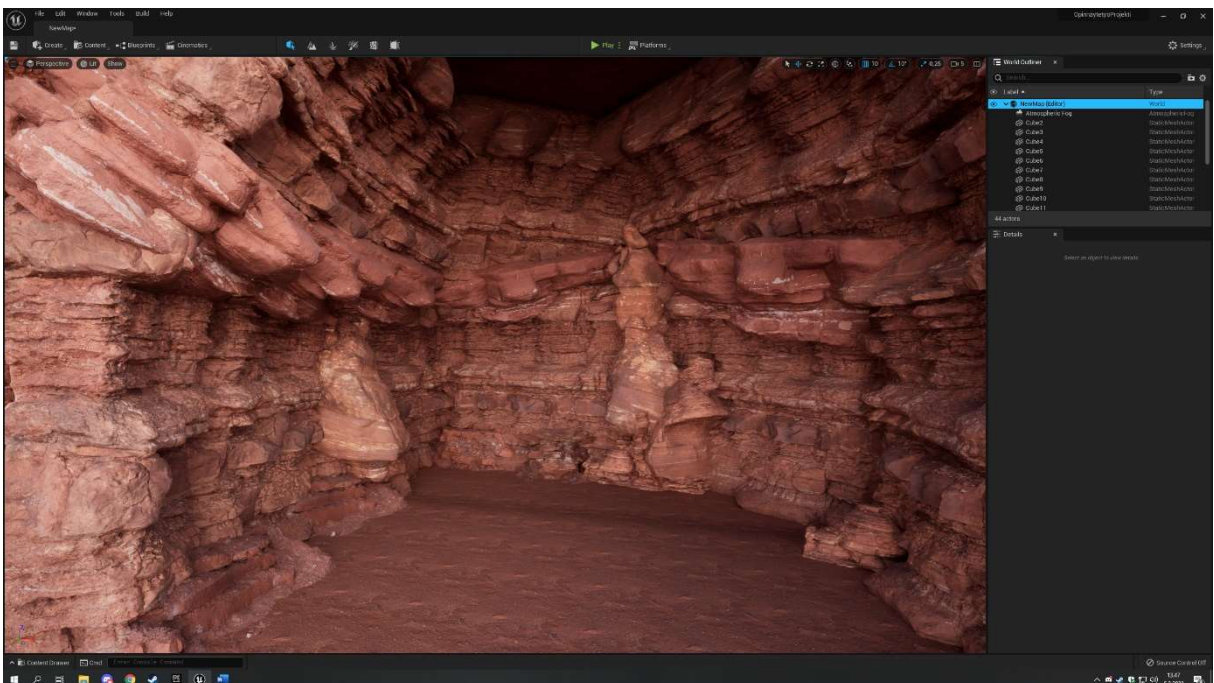
Editori tarjoaa näkymän, jossa voidaan visualisoida Naniten cluster-osiot, jolloin nähdään mitkä alueet malleista on ladattu tarkasti. Nanite clusters Näkymän saa päälle yläpalkin alapuolella olevasta view moden alaluettelosta Nanite visualization. Kuvassa 10 huomaa, kuinka paljon enemmän clustereita, täten myös yksityiskohtia Nanite-valmis malli sisältää. Jos Nanitea haluaa käyttää, kannattaa suosia valmiita nanitelle tehtyjä malleja, jos ei muuten löydy samantasoisia.

Kuva 10. Nanite Visualization. Vasemmalla itse luotu Nanite. Oikealla Nanite-valmis malli. Mitä suurempi määrä clustereita, sitä tarkempi malli.



Naniten demonstroinnin jälkeen rakennetaan pieni taso Nanite-malleista käyttäen samaa Canyons of Utah -kirjastoa. Kuvassa 11 on rakennettu luola käyttäen Quixel bridgestä ladattuja Nanitea käyttäviä Canyons of Utah -malleja. Luola rakentuu useista erimuotoisista ja -kokoisista palasista.

Kuva 11. Nanite-malleista rakennettu taso.

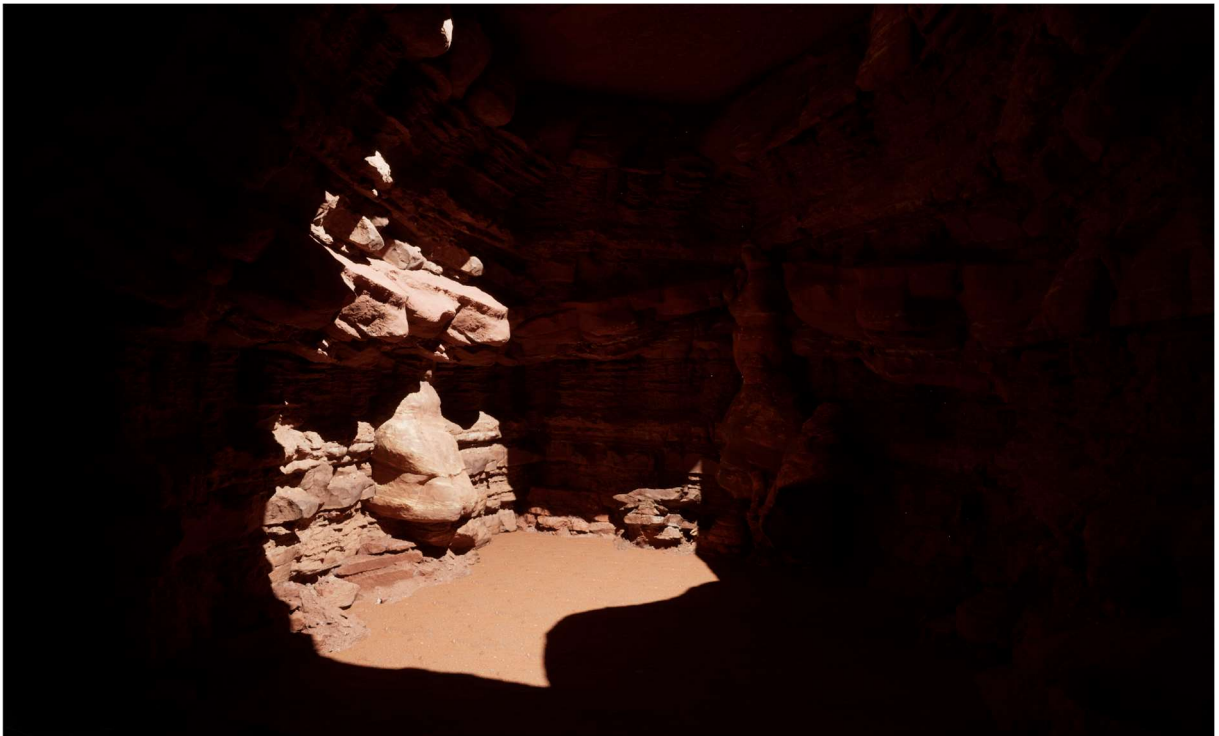


6.4 Lumen

Lumen on aina automaattisesti päällä kaikissa uusissa projekteissa Unreal Engine 5:ssä, mutta ei täysillä asetuksilla. Tämän projektin luonnissa asetettiin päälle laitteistotason säteenseuranta. Rakennettu taso on suljettu luola, jonka katossa on reikä. Reiän päälle asetetaan valonlähde, joka toimii kuin aurinkona ja valaisee kirkkaasti vain osan luolan lattiasta ja seinästä. Tasossa on SkyLight esine, joka lisää pelitason joka paikkaan passiivisesti hieman valoa. Otetaan tämä SkyLight pois päältä painamalla esinelistassa sen kohdalla silmän kuvaa, jotta Lumenin epäsuoran valon näkisi paremmin.

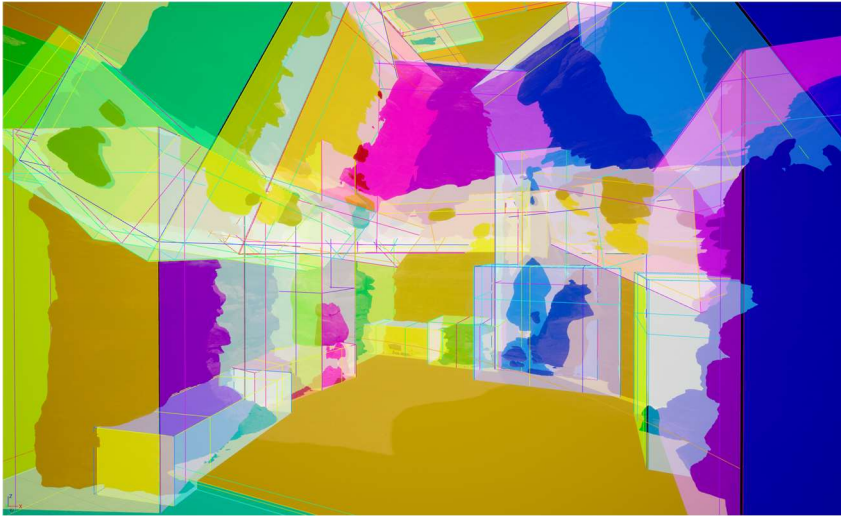
Kun tasosta on saatu tarpeeksi pimeä piilottamalla SkyLight, katon aukosta paistava valo heijastuu lattiasta ja seinistä muuhun ympäristöön realistisesti valaisten koko luolaa. (Kuva 12)

Kuva 12. Lumenilla epäsuorasti valaistu luola.



Syöttämällä editorin alareunan konsoliin komento `r.Lumen.Visualize.CardPlacement 1`, voidaan nähdä kaikki Lumenin käyttämät kortit epäsuoran valaistuksen laskelmointiin, joita alkaa olemaan aika paljon tällaisessa tasossa. (Kuva 13)

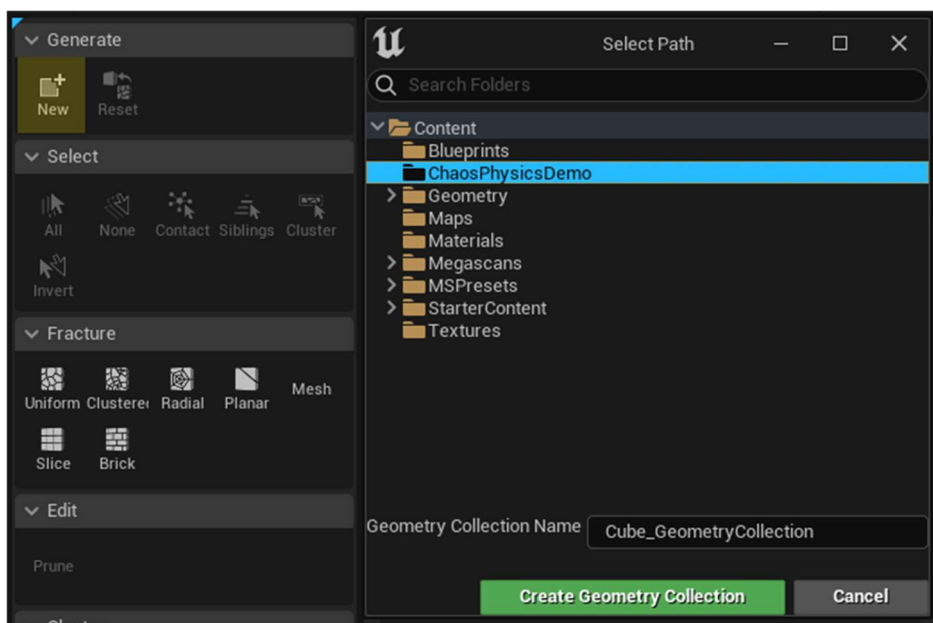
Kuva 13. Tason Lumen Kortit.



6.5 Chaos Physics

Unreal Engine 5:ssä voidaan luoda editorin sisällä mistä vain esineestä tuhottava käyttämällä Fracturing-, eli murtumatyökalua, joka löytyy editorin yläpalkista. Lisätään tasoon kuutio, tehdään siitä isompi ja kuutio valittuna painetaan yläreunasta Fracture Editing Mode. Painetaan vasemmassa olevasta valikosta New, jotta voidaan luoda uusi kansio murtuman aseteille. Tämä vaatii kansion, joka on tässä projektissa nimeltään ChaosPhysicsDemo. Lopuksi painetaan Create Geometry Collection. Jokaisesta murtuman palasesta tulee oma esineensä. (Kuva 14)

Kuva 14. Fracture Editorissa uuden Fracturen luonti. Keltaisella merkitty painike

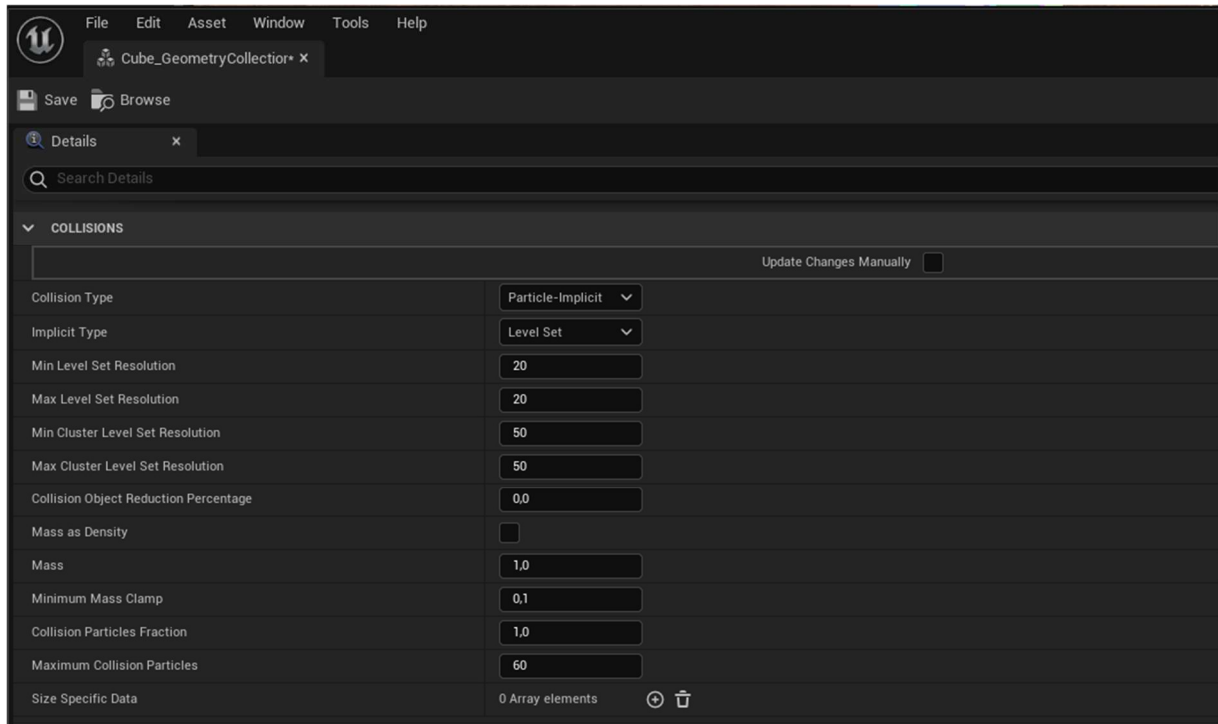


Kun uusi murtuma asset on luotu, tulee editorin listaan uusi Cube_GeometryCollection, joka toimii kokoelmana palasille. Valitaan murtumaeditorista Uniform Fracture ja valitaan sen asetuksista Uniform Voronoi Maximum, jonka arvoksi laitetaan tässä esimerkissä 50. Minimiksi laitetaan 20, jolloin murtumien määrä on satunnainen näiden kahden luvun väliltä. Painamalla Fracture, editori luo palaset esineestä. Murtumilla voi olla eri tasoja, joita voidaan lisätä painamalla Fracture-painiketta useasti. Näin ensimmäisen tason palasia voidaan edelleen hajottaa vielä pienempiin osiin.

Oletuksena, luoduilla palasilla on liian isot box-colliderit, mikä saa esineen hajoamisen näyttämään räjähtävältä ja leijuvalta esineen murtuessa syntyvien box colliderien sisäkkäin menon takia. Valitaan esine ja tuplaklikataan asetuksista Chaos Physics osiosta Rest Collection kohtaa. Uudesta valikosta, voidaan valita colliderin muoto Implicit Type -luettelopainikkeesta. Box-colliderit ovat suorituskyvyn kannalta kevyitä ja niiden kokoa voidaan muuttaa vaihtamalla Collision Object Reduction Percentagea. Asettamalla pienempi luku, esimerkiksi 50, colliderien koko puolittuu, jolloin palaset putoavat kasaan luonnollisesti. Colliderin muodoksi voi myös vaihtaa pallon tai kapselin, mutta tarkin vaihtoehto on Level set, joka jäljittelee itse mallin muotoa, mutta on huomattavasti raskaampi prosessoida. Level setille on useita asetuksia sen optimoimiseksi. Käytettäessä

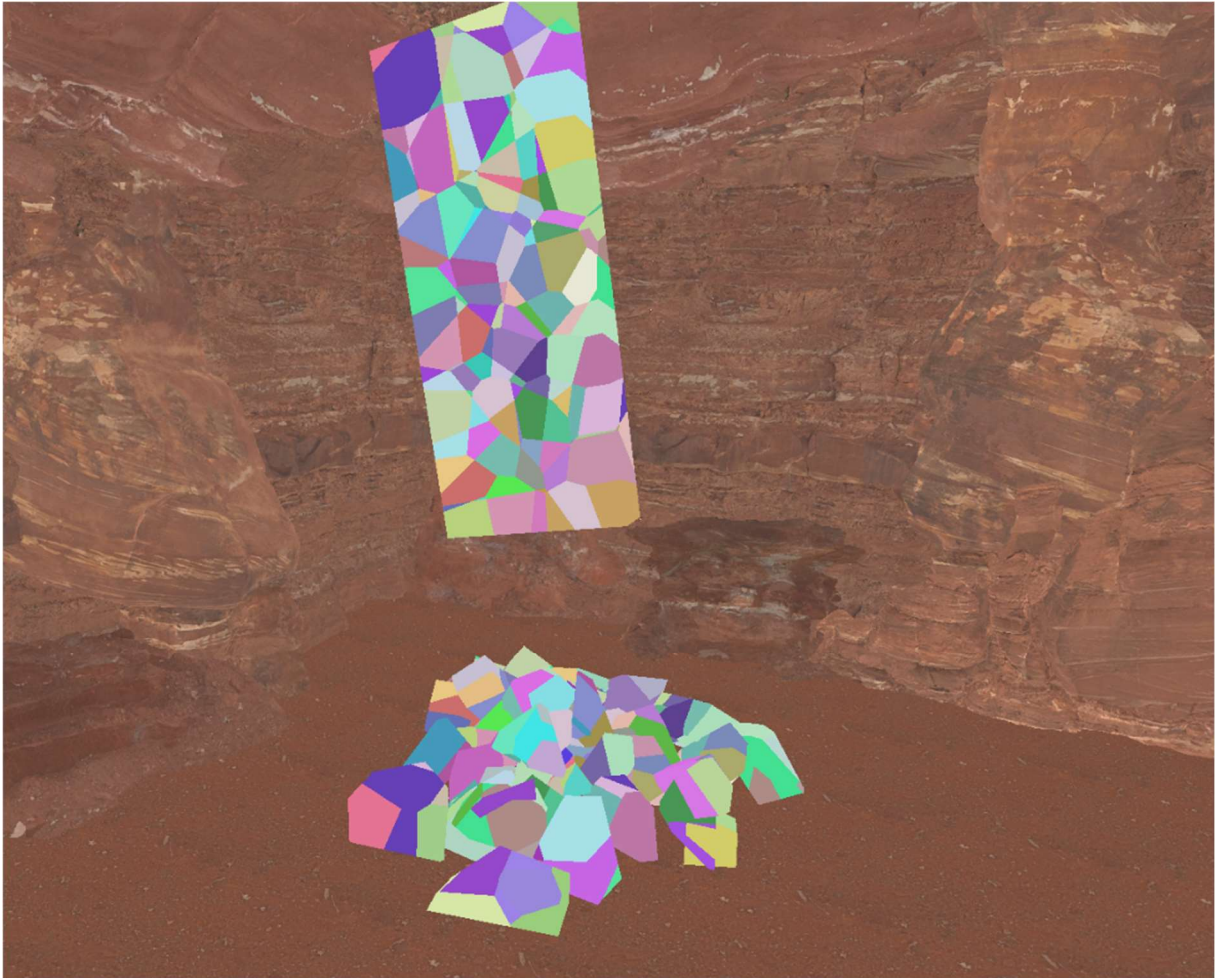
level set collidereita on myös suositeltavaa käyttää Collision Typenä Particle-Implicit. Tässä projektissa tullaan käyttämään level set collidereita. (Kuva 15)

Kuva 15. Murtuvan esineen collider asetukset.



Unreal Engineissä voidaan simuloida peliä valitsemalla ylhäällä olevasta Play-valikosta Simulate tai pikanäppäinyhdistelmällä Alt+S. Tämä ominaisuus on hyvä fysiikan kokeilemiseen, sillä kamera ei hyppää pelaajan näkymään vaan pysyy editorinäkymässä. Valmista murennettua kuutiota testataan ja se hajoaa osuessaan maahan kuvassa 16.

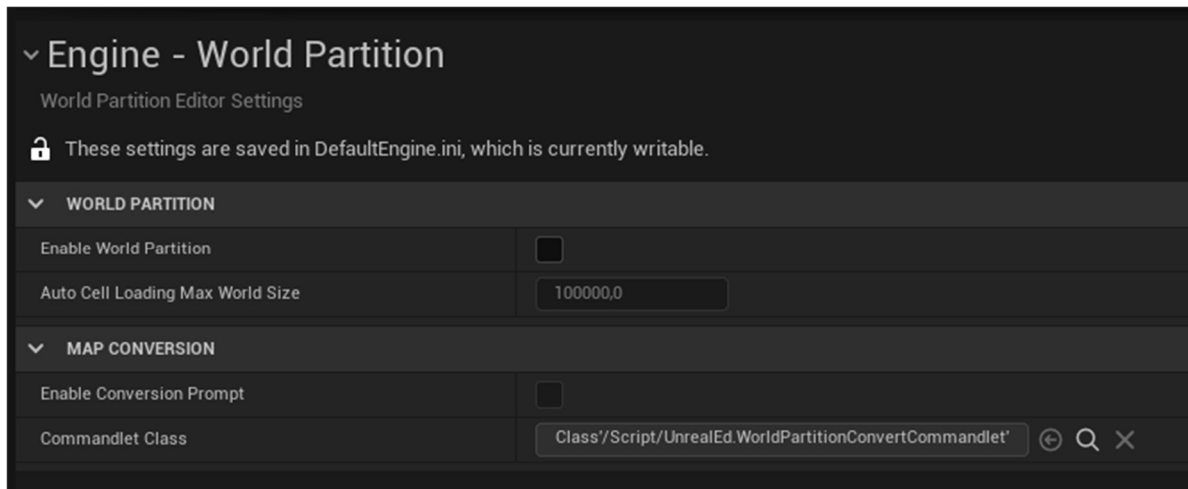
Kuva 16. Kuva, jossa yhdistettynä näkyy kuutio ennen ja jälkeen pelin simuloinnin.



6.6 World Partition

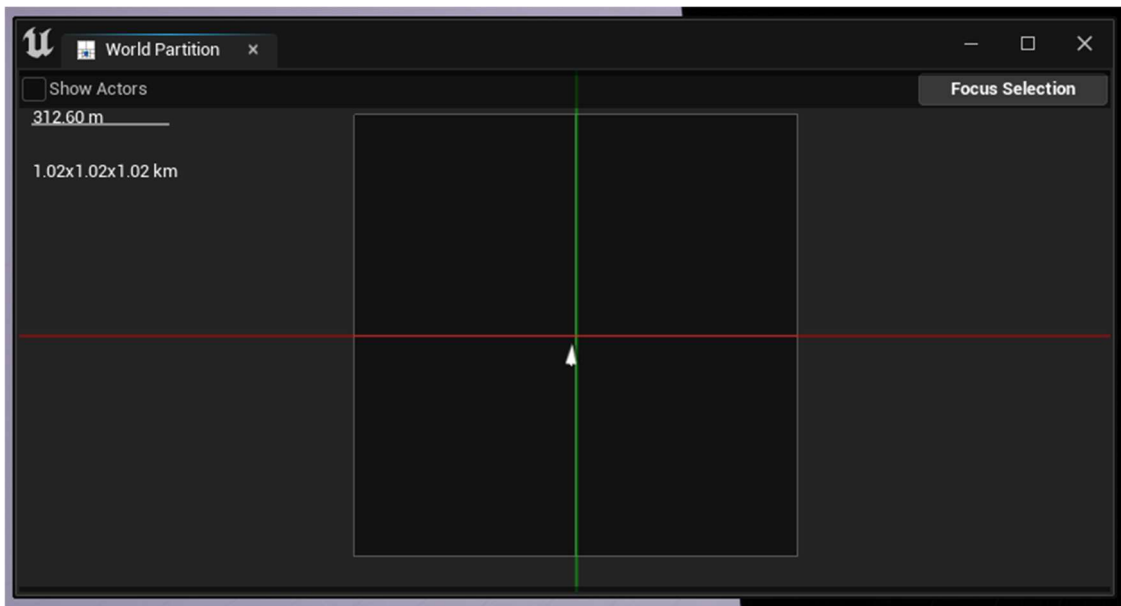
World Partition on hyödyllinen suurien tasojen käsittelyssä. Ominaisuus voidaan laittaa päälle projektiasetusten World Partition sivulta (kuva 17). Vain asetuksen päällelaiton jälkeen luoduissa tasoissa on world partition päällä. Jo olemassa olevat tasot voidaan myös muuntaa world partition map conversion commandletilla windowsin komentokehotteen kautta tai editorista. Muuntamiseen komentokehotteesta löytyy ohjeet Unreal Enginen dokumentaatiosta, mutta tässä esimerkissä muunnetaan käyttämällä editorin sisäistä muuntoa.

Kuva 17. World Partition -projektiasetukset



Kun projektin asetuksista laitetaan päälle world partition ja map conversion prompt, avaamalla tason content vieweristä editori kysyy, haluatko muuntaa tason. Valitaan kyllä ja esiin tulee ikkuna, jossa on valinta In Place, joka tarkoittaa tullaanko taso korvaamaan vai luodaanko siitä uusi versio `_WP` tunnisteella tiedoston nimessä. Tehdään erillinen taso, jos jokin menee pieleen muunnon aikana. Painamalla OK editori suorittaa muunnon ja käynnistyy sen jälkeen uudestaan. Content vieweriin on tullut uusi taso, jonka avatessa se näyttää tyhjältä, mikä on oletettavasti bugi muunnetuissa kartoissa tässä editorin early access versiossa. Ei-ladatut actorit voidaan ladata valitsemalla editorin yläpalkista Window -> World Partition ja oikeaklikkaamalla ruutuja ja valitsemalla Load Selected Cells. Tämä world partition paneeli näyttää kaikki cellit mitä tasossa on ja miten ne latautuvat pelaajan sijainnin mukaan (Kuva 18).

Kuva 18. World Partition -paneeli



7 Demon lopputulos ja huomiot

7.1 Demon lopputulos

Demossa saatiin aikaiseksi haluttu lopputulos, jossa käytettiin editorin uusia ominaisuuksia. Nanite ja Lumen näyttävät hyvältä yhdessä ja suuremmalla määrällä tason rakentamista ja parantelua, esimerkiksi kasvustolla tai monimuotoisemmalla kasvustolla, voitaisiin saada aikaan hyvin realistinen näkymä. Chaos physics myös toimii hyvin ja on helppokäyttöinen. Kyseessä on pieni taso, joka saavuttaa hyvän suorituskyvyn täydellä säteenseurannalla, mutta suurempaa peliä kehittäessä suorituskyky varmasti laskisi ja joitain asioita pitää karsia pois ja optimoinnista tulee tärkeämpää.

7.2 Editorin käyttö

Verrattuna aikaisempaan kokemukseen Unityn kanssa, Unreal Engine vaikutti ensisilmäyksellä paljon käyttäjäystävällisemmältä. Käyttöliittymä on yksinkertaisempi ja muokattavissa. Valikoita on helppo navigoida ja niistä on helppo löytää mitä tarvitsee. Blueprint visuaalisella skriptauksella voidaan tehdä monia asioita ilman ohjelmointia kirjoittamalla.

7.3 Unreal Enginen dokumentaatio

Editorin dokumentaatioon on selvästi käytetty paljon aikaa ja työtä. Dokumentaatio kertoo editorin käytöstä käytännönläheisesti ja helposti ymmärrettävästi. Unrealin dokumentaatio sopii kokemattomille paremmin, verrattuna esimerkiksi Unityn ohjeisiin, joissa ohjeistus on yleensä hyvin teknistä ja ohjelmointipohjaista. Dokumentaation visuaalinen ilme on myös iso osa käyttäjäkokemusta ja Unrealin tapauksessa jokainen ohje on hyvin kuvitettu, eritelty ja sisältää hyödyllisiä sivukommentteja ja neuvoja. Unrealin dokumentaatioissa on myös paljon ohjeita monien perusasioiden tekemiseen alusta alkaen ja työkalujen käyttöön.

8 Yhteenveto

Opinnäytetyön tutkimuskysymyksiin vastattiin onnistuneesti. Ensimmäinen tutkimuskysymys oli, mitä uusia ominaisuuksia Unreal Engine 5 sisältää. Tähän kysymykseen vastattiin esittelemällä pelimoottorin isoimmat ja tärkeimmät uudet ominaisuudet. Toinen kysymys kysyi, miten uudet ominaisuudet toimivat ja miten niitä käytetään. Opinnäytetyössä alkupuolella kerrottiin ominaisuuksien toiminnasta ja demoa tehdessä kerrottiin niiden käytöstä. Viimeinen tutkimuskysymys oli, miten käyttöliittymän muutokset ja parannukset edistävät editorin käyttöä. Käyttöliittymän muutoksista ja parannuksista kerrottiin ja selitettiin miten se vaikuttaa työnkulkuun ja mitä hyötyä niistä on esimerkiksi työpaikoilla, joissa työskennellään tiimeissä.

Opinnäytetyössä Unreal Enginen dokumentaatiosta oli paljon apua tiedonhaussa ja editorin ominaisuuksien toiminnan käsittämiseen ja selittämiseen. Unreal Engine 5 julkaistiin virallisesti opinnäytetyön tekemisen aikana, jossa tuli useita parannuksia ominaisuuksiin, mutta niiden käyttö pysyi samana. Muutamia uusia ominaisuuksia julkaistiin myös, sekä paljon bugikorjauksia. (Epic Games, 2022)

Opin opinnäytetyötä tehdessä paljon Unreal Enginen peruskäytöstä ja kuinka se eroaa esimerkiksi Unitysta. Unreal Enginen graafinen kykenevyys on suuri ja helpommin toteutettavissa verrattuna Unityyn, vaikka silläkin voidaan tehdä näyttäviä pelejä, mutta suuremmalla vaivalla. Unreal Enginen käyttäminen on helppoa, eikä editorin sisäisten työkalujen takia tarvita paljoa ulkopuolisia työkaluja. Unreal Enginen kauppapaikassa ei ole ehkä yhtä paljoa materiaalia verrattuna Unityn asset storeen, mutta laadullisesti se on parempaa, varsinkin kun kyseessä on 3D-skannatut Nanite kirjastot.

Pelimoottorit ja niiden toiminnot ovat niin laajoja, että tällaisesta projektista, joka keskittyi myös täysin uusiin ominaisuuksiin, jäi opittavaa vielä paljon. Jotta voisin tulevaisuudessa oppia lisää Unreal Enginestä, minulla on ideana tutustua moottorin käyttöön paremmin ja tehdä vapaa-ajalla jonkinlainen peli, jossa käytän kokonaisvaltaisesti useampia moottorin ominaisuuksia hyväksi.

Lähteet

Toftedahl M. & Engström H. (2019). *A Taxonomy of Game Engines and the Tools that Drive the Industry*. Skövden Yliopisto.

http://www.digra.org/wp-content/uploads/digital-library/DiGRA_2019_paper_164.pdf

Edwards B. (2009). *From The Past To The Future: Tim Sweeney Talks*. Game Developer.

<https://www.gamedeveloper.com/design/from-the-past-to-the-future-tim-sweeney-talks>

Epic Games. (n.d.-a) *Frequently Asked Questions (FAQ)*. Haettu 8. Helmikuuta 2022

<https://www.unrealengine.com/en-US/faq>

Epic Games. (n.d.-b) *Unreal® Engine End User License Agreement For Publishing*. Haettu 8. Helmikuuta 2022

<https://www.unrealengine.com/en-US/eula/publishing>

Epic Games. (n.d.-c) *Unreal Engine 5 Early Access*. Haettu 8. Helmikuuta 2022

<https://www.unrealengine.com/en-US/unreal-engine-5>

Epic Games. (n.d.-d) *Hardware and Software Specifications*. Haettu 8. Helmikuuta 2022

<https://docs.unrealengine.com/4.27/en-US/Basics/InstallingUnrealEngine/RecommendedSpecifications/>

Epic Games. (n.d.-e) *Blueprint Visual Scripting*. Haettu 10. Helmikuuta 2022

<https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/>

Epic Games (n.d.-f) *Actors*. Haettu 10. Helmikuuta 2022

<https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/UnrealArchitecture/Actors/>

Epic Games (2021) *Unreal Engine 5 Early Access Release Notes*. Haettu 11. Helmikuuta 2022

<https://docs.unrealengine.com/5.0/en-US/ReleaseNotes/>

Epic Games (n.d.-g) *Nanite*. Haettu 11. Helmikuuta 2022

<https://docs.unrealengine.com/5.0/en-US/RenderingFeatures/Nanite/>

Epic Games (n.d.-h) *Lumen*. Haettu 12. Helmikuuta 2022

<https://docs.unrealengine.com/5.0/en-US/RenderingFeatures/Lumen/>

Epic Games (n.d.-i) *Lumen Technical Overview*. Haettu 12. Helmikuuta 2022

<https://docs.unrealengine.com/5.0/en-US/RenderingFeatures/Lumen/TechOverview/>

Epic Games (n.d.-j) *World Partition*. Haettu 13. Helmikuuta 2022

<https://docs.unrealengine.com/5.0/en-US/WorldFeatures/WorldPartition/>

Epic Games (n.d.-k) *MetaSounds: The Next Generation Sound Sources*. Haettu 13. Helmikuuta 2022

<https://docs.unrealengine.com/5.0/en-US/metasounds-the-next-generation-sound-sources-in-unreal-engine/>

Epic Games (n.d.-l) *Virtual Shadow Maps*. Haettu 15. Helmikuuta 2022

<https://docs.unrealengine.com/5.0/en-US/RenderingFeatures/VirtualShadowMaps/>

Epic Games (n.d.-m) *Chaos VS PhysX Overview*. Haettu 18. Helmikuuta 2022

<https://docs.unrealengine.com/5.0/en-US/PhysicsFeatures/ChaosVSPhysxOverview/>

Epic Games (n.d.-n) *Unreal Engine 5 Editor Improvements*. Haettu 20. Helmikuuta 2022

<https://docs.unrealengine.com/5.0/en-US/EditorImprovements/>

Epic Games (2022) *Unreal Engine 5.0 Release Notes*. Haettu 28. Huhtikuuta 2022

https://docs.unrealengine.com/5.0/en-US/unreal-engine-5_0-release-notes/

Epic Games (n.d.-o) *Working With Plugins*. Haettu 18. Elokuuta 2022

<https://docs.unrealengine.com/5.0/en-US/working-with-plugins-in-unreal-engine/>

Liite 1: Aineistonhallintasuunnitelma

Ennen projektia ja sen aikana tehdään muistiinpanoja päiväkirjamaisesti, joihin kerätään teknistä tietoa ja ideoita sisällöstä ja sen rakenteesta. Projektista kertyy kuvankaappauksia ja videoita tehdystä työstä. Kuvankaappauksia käytetään opinnäytetyössä ohjeistuksina ja esimerkkeinä, videoita tehdyn työn näyttämiseen ja dokumentointiin.

Aineiston Luonne:

Opinnäytetyön aikana tuotetun aineiston ja tulokset omistaa opinnäytetyön tekijä itse. Aineisto ei sisällä salassa pidettäviä henkilötietoja tai yrityssalaisuuksia. Kolmannen osapuolen aineistoja käytettäessä on varmistettu niiden käyttöoikeudet. Tietoperustana käytetty aineisto merkitään lähdeluetteloon.

Aineiston Säilytys:

Aineistoa ja opinnäytetyötä säilytetään tekijän tietokoneella, sekä useina varmuuskopioina HAMKin OneDrive-pilvipalvelussa. Videoita säilytetään tietokoneella ja YouTube-videopalvelussa yksityisinä videoina. Opinnäytetyö ja osa aineistosta tallennetaan ajoittain Wihi-palveluun. Kaikkea aineistoa säilytetään opinnäytetyön julkaisun jälkeen vuoden ajan.