

Unity 2022.3 ja Unreal Engine 5 simulaattorikäytössä

Pelimoottorien vertailu

Lari Kettunen

OPINNÄYTETYÖ
Toukokuu 2024

Tietojenkäsittelyn tutkinto-ohjelma
Games Production

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittelyn tutkinto-ohjelma
Games Production

KETTUNEN LARI:

Unity 2022.3 ja Unreal Engine 5 simulaattorikäytössä
Pelimoottorien vertailu

Opinnäytetyö 31 sivua, joista liitteitä 0 sivua
Toukokuu 2024

Opinnäytetyössä selvitettiin, olisiko toimeksiantaja Creanex Oy:n kannattavaa siirtyä käyttämään Unreal Engine 5 -pelimoottoria. He käyttävät tällä hetkellä Unity 2022.3 -pelimoottoria. Selvitys tehtiin rakentamalla toisiaan vastaavat skenet molempiin pelimoottoreihin ja mittaamalla niiden tuottama suorituskky erilaisilla asetuksilla. Skenen maastona toimii laskettelukeskus Alpeilla.

Työssä tutkittiin pääasiassa Unreal Engineä ja sen ominaisuuksia, koska Unity on yritykselle jo entuudestaan tuttu. Unreal Engine 5:n toiminnot Nanite ja Lumen ovat erityisen kiinnostavat simulaattoreita ajatellen, joten niitä tarkastellaan hieman lähemmin. Opinnäytetyössä ei tarkasteltu efektejä, animaatioita tai koodin osuutta.

Tulokset mitattiin valmiilla skeneillä käyttäen apuna ohjelmaa, joka tallentaa koko mittauksen aikaisen suorituskyyyn. Simulaattoreissa on useasti monia aktiivisia kameroita, joten myös testit suoritettiin yhdellä ja kolmella kameralla.

Työssä vertailtiin kahden eri pelimoottorin ominaisuuksia ja suorituskkyä. Työn tulokset auttavat Creanex Oy:tä hahmottamaan pelimoottorin vaihtoon liittyviä haasteita.

Asiasanat: skene, suorituskky, vertautua, simulaattori

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Games Production

KETTUNEN LARI:

Unity 2022.3 and Unreal Engine 5 in Simulator Use
Game Engine Comparison

Bachelor's thesis 31 pages, appendices 0 pages
May 2024

The purpose of this thesis was to help Creanex Oy find out if it is reasonable to change their current Unity 2022.3 game engine into Unreal Engine 5. Creanex builds various heavy machinery training and testing simulators. The study was done by creating the same scene in both game engines and then comparing their performance. The area replicated in the scenes is a skiing resort in the Alps of Italy.

The main focus was on Unreal Engine and its features, since Unity was already known by the company. Unreal Engine 5's features Nanite and Lumen are specifically interesting, therefore they were investigated more thoroughly. The study focused mainly on world and asset creation, because the scene has the largest impact on the simulator performance.

The results were captured with a logging software from a ready build of each scene. The most important results were the comparisons on multi-monitor performance and frame pacing, the overall feed smoothness. Conclusions were that Unreal Engine might perform slightly worse than Unity on average, but Unreal Engine provides a smoother experience for its users. Unreal Engine also has the potential for much higher image fidelity.

Key words: scene, compare, performance, simulator

SISÄLLYS

1	JOHDANTO	8
2	PELIMOOTTORI.....	9
	2.1 Mikä on pelimoottori?	9
	2.2 Miksi ne ovat olemassa	9
	2.3 Mitä pelimoottori tekee?	10
3	UNITY	11
	3.1 Käyttötarkoitus	11
	3.2 Millaisille kehittäjille suunnattu	11
	3.3 Yhteisö	12
	3.4 Hinnoittelu	12
4	UNREAL ENGINE 5.....	14
	4.1 Käyttötarkoitus	14
	4.2 Nanite.....	14
	4.3 Lumen	15
	4.4 Hinnoittelu	15
5	CREANEXIN SIMULAATTORI	16
	5.1 Millainen tuote on kyseessä	16
	5.2 Simulaattorien skenet.....	16
	5.3 Suorituskyvyn haasteet	17
6	3D-MALLIEN LUOMINEN.....	18
	6.1 Työkoneiden mallit	18
	6.2 Maastot	18
	6.3 Rekvisiitta.....	19
	6.4 Optimointi.....	19
7	PROJEKTIN MAASTO.....	21
	7.1 Laskettelurinne.....	21
	7.2 Ympäröivä maasto	22
	7.3 Tarvittavat assetit	22
	7.4 Haasteet.....	22
8	UNITY SKENE.....	25
	8.1 Skenen rakentaminen	25
	8.2 Optimoinnit.....	26
9	UNREAL ENGINE 5 SKENE	27
	9.1 Skenen rakentaminen	27
	9.2 Haasteet.....	28
	9.3 Optimoinnit.....	28

10 MITTAUKSET	30
10.1 Miten mittaukset tehdään.....	30
10.2 Mitattavat skenaariot.....	30
11 TULOKSET	32
11.1 Unity vs Unreal Engine 5	32
11.2 Lumen ja Nanite	33
11.3 UE5 reunanpehmennys	36
11.4 Onko pelimoottorin vaihto hyvä idea?.....	37
12 POHDINTA	38
LÄHTEET	39

LYHENTEET JA TERMIT

AR	Augmented Reality, eli lisätty todellisuus on tekniikka, jossa kamera kuvaa käyttäjän näkemää ja lisää virtuaalisia asioita lopulliseen kuvaan, jonka käyttäjä näkee.
Asset	Termi yksittäiselle tekijän luomalle resurssille, esimerkiksi 3D-malli tai tekstuuri.
Dollari	Amerikan dollari, USD.
FPS	Frames Per Second, eli ruudunpäivitysnopeus. Kuinka monta kuvaa näytönohjain piirtää sekunnissa.
Frustrum Culling	Jättää väliin kameran näkökentän ulkopuolella olevien objektien piirtämisen.
Instansointi	Tapa monistaa objekteja, jolloin ne prosessoidaan vain kerran, eikä yksittäin jokaista erikseen.
LOD, LOD-taso	Level Of Detail tarkoittaa pelimoottorin toimintoa, joka näyttää eri tarkkuisia saman esineen malleja katseletäisyyden mukaan. Kauempana malli näyttää samalta vaikkei olisi yhtä tarkka. LOD-taso tarkoittaa yhtä porrasta näistä malleista.
Lumen	UE5-toiminto, joka jäljittelee säteenseurannan tuottamaa valaistusta ja sen käyttäytymistä.
Malli	3D-malli.
Materiaali	Kokoelma tekstuureita ja arvoja, jotka yhdessä tarkemmin jäljittelevät haluttua pintaa, kuin pelkkä tekstuuri.
Nanite	UE5-toiminto, joka luo annetusta 3D-mallista version, jonka tarkkuutta pelimoottori osaa muuttaa sen näkyvän alueen mukaan.
Occlusion Culling	Jättää väliin kameran näkökentässä piilossa olevien objektien piirtämisen.
PBR	Physically Based Rendering. Tapa, joka pyrkii luonnollisesti jäljittelemään valon käyttäytymistä erilaisilla materiaalinnoilla.
Reunanpehmennys	Keino vähentää näytölle piirrettyjen objektien reunojen porrastusefektiä.

Shader	Eräänlaiset ohjeet näytönohjaimelle, kuinka resurssi piirretään näytölle.
Skene	Avaruus, johon pelin tila rakennetaan.
Tekstuuri	Kuva, johon on määritelty pinnan ominaisuuksia, kuten väri, pinnan muoto tai heijastuksen määrä.
UE5, Unreal	Unreal Engine 5 -pelimoottori.
Visuaali	Creanexin käyttämä termi simulaattorisovelluksen käyttäjälle näkyvästä osasta.
VR	Virtual Reality, eli virtuaalitodellisuus toteutetaan koko katseen peittäväillä laseilla, jotka näyttävät virtuaalisen ympäristön oikeasti kolmiulotteisena ja kamera seuraa sensoreiden avulla pään liikkeitä.

1 JOHDANTO

Toimeksiantaja Creanex Oy haluaa selvittää Unreal Engine 5:n kannattavuutta verrattuna nykyisellään käytössä olevaan Unityn 2022.3-versioon. Creanex toteuttaa asiakkailleen monenlaisia työkonesimulaattoreita erilaisiin tarpeisiin. Pelimoottorin valinnalla on suuri merkitys siihen, millaisia asioita simulaattoreissa voidaan toteuttaa ja millainen on käyttäjän visuaalinen kokemus.

Tämän opinnäytetyön tavoite on selvittää artistin näkökulmasta mitä kaikkea prosessissa ja lopputuloksessa muuttuu, kun pelimoottori vaihdetaan. Selvityksen piiriin kuuluvat pääasiassa objektien mallinnus, maailman luominen ja sen optimointi. Unreal Engine 5:n uudet ominaisuudet Nanite ja Lumen ovat erityisen tarkassa tarkastelussa, sillä varsinkin Nanite saattaa merkittävästi vaikuttaa työkoneiden mallien luontiprosessiin ja mahdollisiin simulaattoriharjoituksiin.

2 PELIMOOTTORI

2.1 Mikä on pelimoottori?

Pelimoottorin on tarkoitus olla pohja, jonka päälle pelin voi rakentaa. Yksinkertaistettuna se on IDE, integroitu kehitysympäristö, joka on varta vasten tarkoitettu pelien tuottamiseen. Pelien tuottamiseen ei varsinaisesti tarvita muuta kuin skriptieditori ja koodin kääntäjä kuten esimerkiksi suositussa ohjelmointiin käytetyssä Microsoft Visual Studiossa on, mutta tässä tapauksessa aivan kaikki näytön piirtämisestä monimutkaisiin toiminnallisuuksiin pitäisi kirjoittaa itse. (Buttle 2020).

2.2 Miksi ne ovat olemassa

Pelien aikojen alussa ei ollut varsinaisia pelimoottoreita, vaan koko peli kirjoitettiin alusta loppuun tukemaan vain sitä laitetta, jolle se julkaistiin. Jos peli julkaistiin useammalle laitteella, kirjoitettiin se tällöin aina uudelleen toimimaan halutuilla laitteilla. (Buttle 2020).

Pelimoottori erottaa monikäyttöisen toiminnallisuuden ja yleensä kertakäyttöisen pelin maailman toisistaan. Tällöin kehittäjät pystyvät käytännössä luomaan uuden pelin samalle pohjalle pienemmillä muutoksilla, tehden prosessista nopeampaa ja edullisempaa. Ideaalinen pelimoottori olisi sellainen, johon ei koskaan tarvitsisi tehdä muutoksia. Tällöin se sisältäisi aina kaiken tarvittavan toiminnallisuuden mitä tahansa peliä ajatellen. Tähän ei luultavasti koskaan päästä, koska tällaista tarvittavaa toiminnallisuutta on lähes ääretön määrä. Tämän vuoksi pelimoottorit kehitetään tai sovelletaan sopimaan erityisen hyvin jotakin tiettyä tarkoitusta varten. Mitä monikäyttöisempi pelimoottori on, sitä heikompi se yleensä on kuhunkin käyttötarkoitukseen erilaisten kompromissien vuoksi. (Gregory 2018).

2.3 Mitä pelimoottori tekee?

Pelimoottori on kokoelma toimintoja, jotka tukevat pelin kehittämistä. Näitä toimintoja tyypillisesti ovat skriptien käsittely, kuvan renderöinti, animaatioiden toistaminen, tekoäly, fysiikat, äänet, internet toiminnallisuudet, editorit ja syötteen hallinta. Skriptit ovat yksinkertaista koodia, jotka ohjaavat peleissä tapahtuvia objektien toimintoja, kuten oven avaaminen. Renderöinti tuottaa pelin kuvan ruudulle. Animaatiot muokkaavat tai liikuttelevat objekteja. Tekoäly ohjaa reitinetsintää ja reagoi ympäristöön. Fysiikat tuovat mukanaan esimerkiksi objektien törmäykset, massan ja painovoiman. Äänet voivat olla kaikkea yksittäisestä äänitiedostosta monien äänien sekoitukseen ja efekteihin. Internetin toiminnallisuuksiin kuuluu yleensä moninpeli, pelin sisäiset ostokset ja tiedonkeruu. Editoreilla rakennetaan pelimaailma ja muokataan asetteja. Syötteen hallinta kääntää käyttäjän moninaisten syötelaiteiden signaalit pelin luettavaksi. (Cowan & Kapralos 2014).

3 UNITY

3.1 Käyttötarkoitus

Unityn suunnittelun pohjana on pelituotannon helpottaminen ja tuki useille eri alustoille, eli laitteille ja käyttöjärjestelmille. Editorilla pystyy helposti tekemään muutoksia pelimaailmaan ja asetteihin, sekä myös testaamaan keskeneräistä projektia suoraan editorissa, tai vaikka kohdealustalla. Unity tarjoaa hyvät perustyökalut melkeinpä minkä tahansa pelin kehittämiseen. (Gregory 2018).

Unity on suosittu valinta, kun kohteena on mobiililaitteet, tai peli on tarkoitus julkaista monilla eritehoisilla alustoilla. Periaatteessa eri alustoille julkaisu on niinkin yksinkertaista kuin valintalaatikon rastittaminen, mutta vähempisuorituskykyisille alustoille täytyy muistaa tehdä asetteja, jotka eivät ole liian raskaita suorittaa. (Buttfield-Addison 2023).

Jos pelin ominaisuuksiin kuuluu jotakin ennennäkemätöntä ja outoa, on Unity silloin Unreal Engineä parempi vaihtoehto. Unityn laaja muokattavuus skripteillä mahdollistaa sellaistenkin toiminnallisuuksien toteuttamisen, joita varten perinteisesti tarvittaisiin kokonaan uusi pelimoottori. (Buttfield-Addison 2023).

3.2 Millaisille kehittäjille suunnattu

Unity on tarkoitettu kaikille kehittäjille, oli projekti melkeinpä mitä tahansa. Noin kaksi kolmesta tänä päivänä julkaistusta pelistä on tehty käyttämällä Unityä. Unity on erityisen suosittu uusien tai pienten pelistudioiden keskuudessa, todennäköisesti sen helppokäyttöisyyden ja laajasti saatavilla olevan ohjemateriaalin ansiosta. (Buttfield-Addison 2023).

Unityllä voi tehdä tuotteen lähes mille tahansa alustalle. Se tukee mobiililaitteita, konsoleita, tietokoneita ja myös selaimia. 2D, 3D, AR ja VR käytön lisäksi Unityllä voi renderöidä videoita. (Unity Technologies 2024 b). Sitä on myös käytetty auton käyttöliittymän toteutukseen (Dawis & Burrows 2022).

3.3 Yhteisö

Unityn suosion ja suuren käyttäjämäärän avulla sen kauppapaikka on täynnä hyödyllisiä työkaluja, oppaita ja asetteja kaikille helposti saatavilla. Erityisesti erilaisia oppaita on valtava määrä, valtaosa on ilmaisia, mutta samalla niiden laatu vaihtelee hyvin paljon. (Buttfield-Addison 2023).

Koska Unityä käyttävät jatkuvasti kaikenlaiset henkilöt veteraaneista vasta-aloitteijoihin, on saatavilla oleva tiedon ja opastuksen spektri myös laaja. Unityllä on kattava dokumentaatio, mutta se ei aina ole kovin ymmärrettävä. Dokumentaation lisäksi Unityllä itsellään on useita harjoituskursseja, joiden mukana voi itse tekemällä oppia. (Buttfield-Addison 2023).

3.4 Hinnoittelu

Unitystä on muutama erilainen paketti, josta tilaaja maksaa. Henkilökohtainen lisenssi on tällä hetkellä ilmainen siihen asti, kunnes Unity-tuotteita on myyty 100 000 dollarin edestä kuluneen 12 kk aikana. Tämän jälkeen täytyy käyttää maksullisia lisenssejä. Isommille projekteille löytyy Unity Pro, Enterprise ja Industry -paketit, joilla saa pelimoottorin käyttöoikeuden lisäksi useita erilaisia kehitystä helpottavia lisukkeita. Näiden lisenssien hinnat ovat vähän kaikkea vajaan kahden ja viiden tuhannen euron väliltä per vuosi. Personal-lisenssin tulorajaa ollaan parasta aikaa nostamassa 200 000 dollariin vuodessa. (Unity Technologies 2024 a & c).

Syksyllä 2023 Unity Technologies ilmoitti hinnoittelunsa muutoksista, jotka olivat hyvin radikaaleja. Lyhyesti: kehittäjän tulisi maksaa yritykselle jokaisesta tuotteensa asennuskerrasta, välittämättä sen rahoitusmallista tai käyttötarkoituksesta. Tämä tarkoittaisi sitä, että myös ilmaisista sovelluksista ja vaikkapa piraattikopioista tulisi myös maksaa, vaikkei niistä saada tuottoa. Hinnoittelun muutoksista nousi suuri vastustus ja Unity Technologies uudisti nopeasti hinnoittelumalliaan kehittäjille ystävällisempään suuntaan, jolloin asennuskerroista maksettaisi-

siin vasta miljoonan asennuksen ja miljoonan dollarin voiton jälkeen. Tällä sotkulla yritys menetti monien kehittäjien luottamuksen ja aiheutti siirtymää toisiin pelimoottoreihin. (Carpenter 2023 a & b).

4 UNREAL ENGINE 5

4.1 Käyttötarkoitus

Unreal Engine luotiin aikoinaan pohjaksi kehittäjänsä Epic Gamesin Unreal-pelisarjalle. Tästä sitä laajennettiin vuosi vuodelta, kunnes lopulta päädyttiin nykyiseen versioon. Tänä päivänä Unreal Engine on yksi käytetyimmistä ja parhaimmat työkalut omaavista pelimoottoreista. Unreal Enginen erikoisuus on sen visuaalinen blueprint-ohjelmointitoiminto, joka mahdollistaa logiikan ohjelmointia kirjoittamatta koodia. (Gregory 2018).

Unityn tapaan Unreal Enginellä voi luoda oikeastaan minkä tahansa pelin, mutta se voi vaatia pelimoottorin tuunausta omiin tarpeisiin. Muokattavuutensa lisäksi UE:lla on myös laaja seuraajakunta, joten ohjeita ja valmista sisältöä löytyy oikeastaan aiheesta kuin aiheesta. Se siis sopii oikeastaan kenelle tahansa. (Gregory 2018).

Pelimoottorilla on myös oma kauppapaikkansa, josta valmiita asetteja tai työkaluja voi hankkia. Kaupan lisäksi Unreal Enginessä on sisäänrakennettu Quixel bridge, joka mahdollistaa lukemattomien korkeatasoisten megascans-asettien lataamisen maksutta käyttöönsä, kunhan se pysyy pelimoottorin sisällä. (Unreal Engine a).

Unreal Engineä on käytetty monissa paikoissa, mitä ei välttämättä ole edes huomannut. Pelien lisäksi sitä on käytetty esimerkiksi Mandalorian-sarjan kuvauksissa visualisoimaan kuvausten taustoja (Good 2020) ja Rivian R1T -sähköauton käyttöjärjestelmän visualisoinnissa (Unreal Engine b).

4.2 Nanite

Nanite on Unreal Engine 5:n uusi toiminto, joka muokkaa 3D-mallien geometriaa suorituksen aikana. Lyhyesti kuvailtuna Nanite näyttää hyvin läheltä tarkasteltuna

3D-mallin kaiken saatavilla olevan laadun, kun taas kaukaa katsottuna mallin laadua heikennetään lineaarisesti etäisyyden mukaan. Malli siis näyttää aina samalta katsojan silmään, mutta kauempana ollessaan se vie vähemmän suorituskykyä. (Venter & Ogterop 2022).

Tällä hetkellä Nanite soveltuu ainoastaan ”koviin” malleihin, eli sellaisiin, jotka eivät taivu. Parhaillaan se on rakennusten, kivien ja vastaavien kanssa. Nanitea voi käyttää myös puihin ja muuhun kasvillisuuteen, mutta niistä saadut hyödyt eivät ole yhtä suuret. (Venter & Ogterop 2022).

4.3 Lumen

Naniten tapaan Lumen on myös täysin uusi toiminto Unreal Engine 5:ssä. Se on uusi tapa toteuttaa maailman valaistuksen ja heijastusten laskenta. Lumen toimii käyttäen kevyttä säteenseurantaa, jolla se aluksi kerää karkean kuvan valaistuksesta. Säteenseurannan datan ja 3D-mallien muotojen pohjalta luodaan tulos valaistuksesta, joka näytetään käyttäjälle. (Wright, D. & Narkowicz 2024).

Lumen voi käsitellä äärettömän määrän heijastuksia, joka mahdollistaa skenen valaistumisen luonnollisesti heijastuksilla. Tämä tarkoittaa, että perinteisesti käytettyjä näkymättömiä valonlähteitä ei tarvitse käyttää luonnollisen valaistuksen aikaansaamiseksi. (Wright, D. & Narkowicz 2024).

4.4 Hinnoittelu

Unreal Enginen hinnoittelu on samankaltainen Unityn kanssa. Pienissä projekteissa sen käyttäminen on ilmaista ja siitä joutuu maksamaan vasta, kun odotettu tuotto on yli miljoona dollaria. (Unreal Engine 2024).

Isompiin projekteihin on räätälöity hinnoittelu, mutta sitä ei ole virallisilla sivuilla mainittu. Hinnan voi odottaa olevan vastaava Unityn kanssa, noin kahdesta tuhannesta dollarista ylöspäin. (Unreal Engine 2024).

5 CREANEXIN SIMULAATTORI

5.1 Millainen tuote on kyseessä

Creanexin simulaattorit ovat yrityksille tarkoitettuja työkoneiden kehitys- ja koulutussimulaattoreita. Tämä työ koskettaa pääasiassa koulutussimulaattoreita, koska niissä maastot ovat suuremmassa roolissa. Koulutussimulaattorit koostuvat istuimesta, ohjainlaitteista, tietokoneesta ja näytöistä. Joskus niissä on myös liikealusta, joka fyysisesti heiluttelee simulaattoria sen tapahtumien mukaisesti.

Itse tietokonesovellusta ajatellen simulaattorit koostuvat kahdesta osasta: käyttäjälle näkymättömästä fysiikkasimulaatiosta ja näkyvästä visuaalista. Simulaatio kertoo visuaalille mitä sen kuuluu näyttää ja missä, eli pelimoottorin tarjoamia fysiikoita ei käytetä ollenkaan.

Visuaalina toimii tällä hetkellä Unity 2020.3 -versio, mutta vanhemmissa simulaattoreissa on käytetty myös Ogre 2 -moottoria. Nyt tutkinnan alla on siirtyminen Unreal Engine 5:een, jonka Nanite-ominaisuus voisi mahdollistaa paremman lopputuloksen simulaattoreissa esiintyvien työkoneiden tarkkuudessa menettämättä suorituskykyä.

5.2 Simulaattorien skenet

Simulaattorit jäljittelevät todellisuutta. Virtuaalisen maailman tulee olla uskottava ja tärkeiden asioiden mittasuhteet täytyy olla tarkalleen oikein. Jos vaikka rekka-simulaattorissa risteyksien leveys on puoli metriä liikaa, on kääntyminen trailerin kanssa merkittävästi helpompaa, kuin se olisi oikeasti.

Maastot pyritään pitämään kompakteina, jotta suorituskyvyn kanssa ei tulisi ongelmia, mutta esimerkiksi ajosimulaattoreissa maastolla on pakko olla enemmän kokoa. Suurten maastojen kanssa on pakko tehdä kompromisseja objektien tiheydessä ja tarkkuudessa.

5.3 Suorituskyvyn haasteet

Simulaattoreissa käytetään useasti kolmea monitoria, joka Unityssä aiheuttaa suuren ruudunpäivitysnopeuden pudotuksen. Jokaiselle monitorille on oma kamera, ja jokainen ylimääräinen kamera lisää kuormaa näytönohjaimelle. Myös simuloitavien koneiden peileillä on omat kamerat ja peilejä voi helposti olla viisi tai enemmän.

Työkoneiden mallit ovat huomattavasti tarkempia, kuin mihin videopeleissä on totuttu. Suuret kolmiomäärät ja monet isot tekstuurit aiheuttavat enemmän kuormaa, kuin on tyypillistä. Jokainen liikkuva osa on myös erillinen mallinsa ja näitä on joissakin koneissa kymmeniä.

Suuri piirtoetäisyys ja samalla suuret maastot aiheuttavat myös omanlaisensa ongelmat. Käyttäjä ei saa nähdä maailman rajoja, muuten illuusio todellisuudesta menetetään. Maaston muodot, rakennukset ja metsät ovat yleensä ne asiat, jotka rajoittavat näkyvyyttä. Näistä edellä mainituista metsät on vaikein toteuttaa. Metsät ovat luonnossa tiheitä, täynnä erilaista kasvillisuutta, eikä niiden läpi näe. Tällainen tiheys on toki tehtävissä, mutta hyvin suurella suorituskyvyn hinnalla. Kun suorituskykyä ei tällaista varten ole jäljellä, on tehtävä iso kompromissi tiheyden, kasvillisuuden variaation ja niiden laadun väliltä.

6 3D-MALLIEN LUOMINEN

6.1 Työkoneiden mallit

Creanex saa simulaattoreihin tulevien työkoneiden mallit asiakasyrityksiltä CAD-tiedostoina. Aluksi nämä CAD-tiedostot prosessoidaan sovelluksessa nimeltä Pixyz Studio, joka muuntaa matemaattisen CAD-datan kolmioiksi. Kolmioitu malli siirretään FBX-muodossa 3D-käsittelyohjelmaan, joka tällä hetkellä on Blender.

Blenderissä mallista poistetaan mahdollisimman paljon pintaa, jota käyttäjä ei tule näkemään. Ylimääräisen pinnan poiston jälkeen malleja yksinkertaistetaan, jos se ei vaikuta ulkonäköön tai kestä tolkuttoman kauaa. Enemmän tästä optimointikappaleessa.

Mallit teksturoidaan Substance Painterilla, jolla voi myös lisätä helposti likaa tai pieniä yksityiskohtia, kuten tarroja. Substance Painterista tekstuurit voi helposti tallentaa haluamassaan muodossa. Mallit ja tekstuurit yhdistetään Blenderissä ja siirretään sieltä pelimoottoriin.

6.2 Maastot

Maaston malli saadaan yleensä jonkin muotoisena korkeusdatana, joka käännetään kolmiomuotoon. Käytetyt tekstuurit ovat tyypillisesti PBR-tekstuureita, joilla maastoihin saadaan luonnollinen ulkonäkö.

Maastojen koko pyritään pitämään kompaktina suorituskykyhaasteiden vuoksi, mutta se ei aina ole mahdollista. Erityisesti ajokoulutussimulaattoreissa on oltava riittävästi tilaa ja vaihtelua harjoitella.

Yksi suurimmista haasteista maastojen kanssa ovat niiden rajat. Käyttäjä ei saisi nähdä mihin maailma loppuu, tai muuten immersio kärsii. Peleissä maailman rajat ovat usein veden peitossa tai vuorien takana, mutta koska simulaattoreissa käytetään aitoja maastoja, nämä keinot eivät yleensä ole hyödynnettävissä.

Useimmiten rajat tai harjoittelualueen reunat peitetään riittävän tiheällä metsällä, jottei kauemmas voi nähdä. Yksi perinteinen keino olisi rajoittaa näkyvyyttä sumulla ja piirtoetäisyydellä, mutta tätä nykyä nämä konstit näyttävät visuaalisesti aika kauheilta.

6.3 Rekvisiitta

Rakennukset ja muut objektit mallinnetaan itse, ja niiden määrä on perinteisesti melko pieni. Kaikella on lähes poikkeuksetta oikea esine mallina, jota jäljitellään tekijänoikeuksien puitteissa. Kuten muillakin käytetyillä malleilla, on objektien mitasuhteet oltava todellisen mukaiset.

Tekstuurit tehdään itse samaan tapaan kuten koneillakin. Lopputulos on yleensä PBR-materiaali, jota säädetään yhtenevän lopputuloksen saamiseksi muiden mallien kanssa. Vanhemmat mallit eivät sisällä PBR-materiaaleja.

6.4 Optimointi

Kaikissa malleissa pyritään pitämään turhat kolmiot minimissä. Erityisesti työkoneiden malleja joutuu yleensä siivoamaan ja mallintamaan päälle. Yleisiä ongelmakohtia koneissa ovat letkut, pultit, prikät ja muu pienrauta, sekä osien sisäpinnat. Koska CAD-tiedostoissa on myös materiaalin paksuus mukana, siirtyy tämä tieto myös Blenderiin. Käyttäjä ei näe materiaalin kääntöpuolta, joten ne voi poistaa. Jos ajatellaan esimerkkinä laatikkoa, olisi perinteisessä pelin mallissa 6 sivua, eli ulkopuolet, mutta CAD-mallissa niitä olisi 12. Jokainen pinta piirretään huolimatta siitä näkyvätkö ne vai ei.

Maastolle paras keino säästää suorituskykyä on yksinkertaisesti tehdä mahdollisimman pieni alue, koska siinä ei ole yhtä paljon kolmioita kuin isommassa. Pienen maaston myös mahtuu ja tarvitaan vähemmän objekteja, jolloin kuorma myös siltä tavoin on pienempi.

Muiden mallien kanssa ei oikein voi tehdä muuta, kuin pitää kolmiomäärän alhaisena ja maastossa käytettyjen objektien määrän pienenä. Puut on pakko pitää melko pienilaatuisina, koska muuten skenen kolmiomäärä kasvaa räjähdysmäisesti käytettyjen puiden määrän vuoksi. Puissa käytetään muutamaa LOD-tasoa, jotka melko nopeasti vaihtuvat 2D-tasoiksi. Puissa on kolmiomäärän lisäksi myös toinen haaste, overdraw, eli läpinäkyvyyttä sisältävien tasojen päällekkäisyys. Jos pikselillä on paljon tällaisia tasoja, joudutaan kaikki päällekkäisistä tasoista käsittelemään, vaikkei niiden läpi enää näkyisikään. Tällaiset pikselit ovat hitaampia renderöidä kuin sellaiset, joissa ei ole läpinäkyvyyttä tai sitä on vähän. Lehdet ja neulaset ovat läpinäkyviä 2D-tasoja, ja metsässä näitä tulee helposti todella paljon päällekkäin. LOD-tasot auttavat vähentämään päällekkäisyyttä kauempana kamerasta.

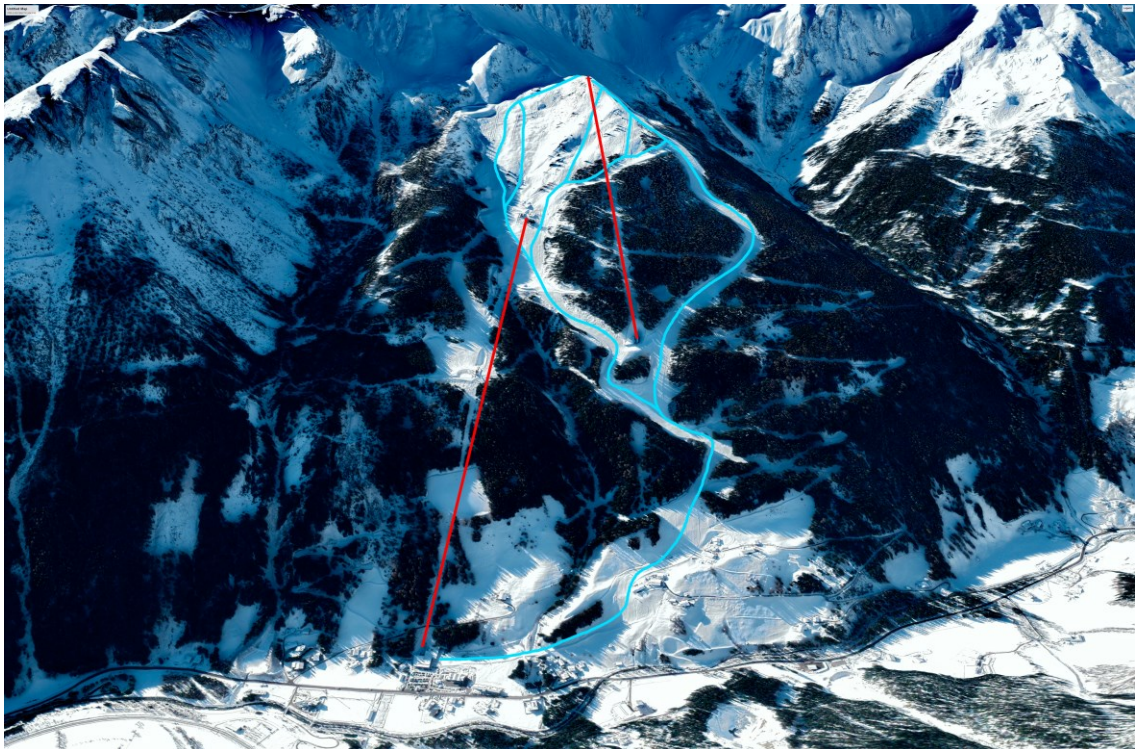
Tekstuurien koko ja materiaalien määrä pyritään pitämään pienenä, mutta erityisesti työkoneisiin niitä tarvitaan yleensä paljon. Jokainen uusi materiaali lisää piirtohäskäjen määrää, mikä myös heikentää suorituskykyä.

7 PROJEKTIN MAASTO

7.1 Laskettelurinne

Opinnäytetyön maastoksi valikoitui Ladurns-laskettelukeskuksen rinteet Italian Alpeilla. Maaston tarkka, laserskannattu malli on saatu erästä yhteistyöhankkeesta. Tämä maasto on tulossa käyttöön lumikissasimulaattorin testausmaastoksi.

Alueen pituus laskettelukeskuksen porteilta rinteiden huipulle on noin 2,7 kilometriä ja korkeutta rinteillä on noin 900 metriä. Alue sijaitsee kahden jyrkän väli-
sessä V:n muotoisessa laaksossa. Alla olevassa kuvassa 1 on havainnollistettu tärkeimmät laskettelukeskuksen rinteet ja hissit. Sisäänkäynti on näkyvillä kuvan alalaidassa vasemmalla puolella merkittyjen viivojen risteyskohdassa.



KUVA 1. Ladurns-laskettelurinteet. Hissit punaisella, päärinteet sinisellä. Lähde: Google Earth.

7.2 Ympäröivä maasto

Koska laskettelurinteen malliin kuului pelkästään laskettelualue, tarvitsi ympäröivä maasto hankkia muualta. Se löytyi OpenDEM-verkkosivulta, josta pystyi lataamaan riittävän tarkkaa korkeusdataa tarvitsemastani alueesta. Korkeusdatan konvertointiin kolmiomuotoon käytettiin Blenderin BlenderGIS-lisäosaa, joka tuotti datan pohjalta maapohjan 3D-mallin oikeassa mittakaavassa.

3D-malli oli sellaisenaan 25 kilometriä pitkä suuntaansa ja siinä oli lähes 10 miljoonaa kolmiota. Valtaosa tästä pinnasta leikattiin pois, koska se jäi kukkuloiden taakse näkymättömiin. Koska maaston tarvitsi ainoastaan näyttää kaukaa hyvältä, sen pystyi sieventämään helposti noin 50 000 kolmioon.

7.3 Tarvittavat assetit

Laskettelurinteen ympäristöön kuuluu maapohjien lisäksi hiihtohissit ja noin 20 rakennusta, joista olennaisimpia ovat laskettelualueen päärakennus, hotelli, hissipysäkkien rakennukset ja ravintolat. Nämä rakennukset on hieman tarkemmin jäljitelty, kuten vaikka pienet alppimökit. Rakennukset näyttävät kauempaa riittävän aidoilta. Muita tärkeitä objekteja ovat muutama erilainen puu ja erilaiset turvaaidat.

Objektien tekstuurit ovat Substance Painterilla luotuja, mutta maapohjaan on käytetty jo ennalta hankittuja tai tätä työtä varten internetistä ladattuja CC0-tekstureita. Unreal Enginen versiossa mukana on myös muutama Quixelista ladattu tekstuuri ja vapaasti saatavilla oleva maapohjan älymateriaali. Näitä hyödynnettäisiin myös silloin, jos UE5-versio olisi tulossa oikeasti käyttöön.

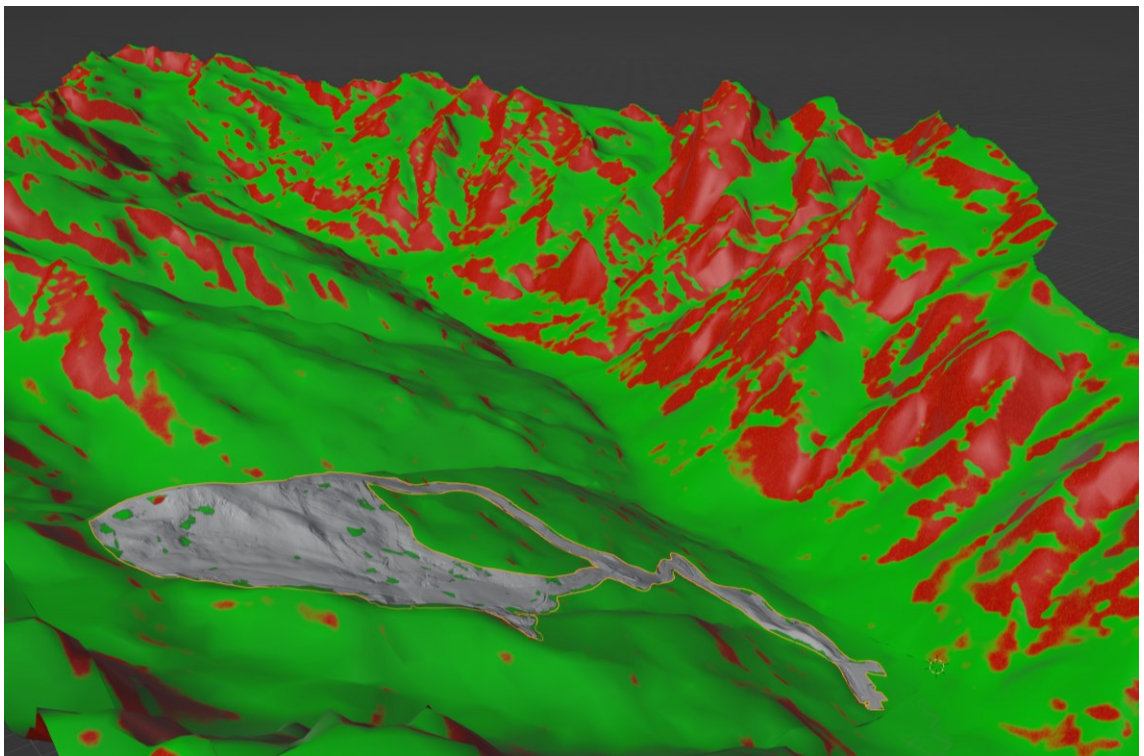
7.4 Haasteet

Yllättävästi maaston massiivinen koko ei ollut suurin haaste projektin asettien luonnissa. Sen sijaan hyvin niukka kuvamateriaali alueen rakennuksista koitui

varsinaiseksi päänvaivaksi, josta myös syntyi päätös jättää kaikki vähemmän tärkeät rakennukset tasaisen karuiksi, ettei muutama erotu ikävästi joukosta.

Toinen ongelma oli rakennusten mittojen ja paikkojen hahmottaminen. Tämä yleensä onnistuu melko vaivatta Google Maps ja Earth -palveluilla. Ikävä kyllä hiihtokeskuksen ohi kulkee vain yksi kamera-auton kuvaama tie ja sen ottama kuvanlaatu jätti toivomisen varaa. Myöskään satelliittien ottamat kuvat eivät olleet suureksi avuksi rinteiden korkeuden ja jyrkkyyden vuoksi, sillä tämä perspektiivi aiheutti suuria paikan siirtymiä rinteiden eri osissa olevien asioiden välillä. Satelliittikuvista pystyi ottamaan rakennusten kattojen mitat, muttei oikein muuta. Rakennusten paikat oli oikeastaan pakko sovittaa suurpiirteisesti kohdilleen.

Pääaluetta ympäröivän maaston suuri koko tarkoitti, että sen teksturoimiseen oli käytettävä jotakin muuta konstia, kuin suoraa tekstuuria. Päädyin käyttämään splat mappia, eli tässä tapauksessa punavihreää tekstuuria, jonka punaisen ja vihreän värikanavan arvot osoittavat pelimoottorille mihin oikeaa materiaalia asetetaan. Kuvassa 2 näkyy käytetty splat-tekstuuri Blenderissä ja kuvassa 3 lopputulos Unityssä.



KUVA 2. Ympäröivän alueen splat-tekstuuri, jossa punainen korvataan kivellä ja vihreä lumella.



KUVA 3. Splat-tekstuuri Unityssä, kun punainen ja vihreä on korvattu oikeilla materiaaleilla.

Pienen kurvipallon aiheutti myös käytössä oleva laskettelurinteiden maapohjan malli. Se on laserskannaus sulasta maasta ja simulaattorissa pitäisi olla talvi, jolloin paksu lumikerros muuttaa merkittävästi rinteiden muotoa. Mallia on muokattu kohottamalla ja tasoittamalla, jotta ne suurin piirtein vastaisivat oikeita, hoidettuja rinteitä.

8 UNITY SKENE

8.1 Skenen rakentaminen

Unityn pelimaailman rakentaminen on jo ennestään tuttua, joten prosessi ei vienyt kovinkaan paljoa aikaa. Maasto koostuu kahdesta palasta, alla olevasta tarkasta maan pinnasta ja etäällä olevasta pienitarkkuisesta taustasta. Tarkan maaston mallin kääntämiseen käytin työkalua, joka muuntaa mallin Unityn maapohjaksi eli terrainiksi. Maapohja oli kannattavaa teksturoida käsin, koska käyttäjän läheltä näkemät pinnat eivät voi olla ihan mitä sattuu. Taustamallille oli luotu splat-tekstuuri jo Blenderissä, joten melkein ainut tehtävä oli laittaa malli paikalleen terrainin ympärille. Splat-tekstuurin käyttöä varten täytyi luoda oma shaderi, joka osaa lukea annetun tekstuurin punaisen ja vihreän kanavan arvot ja näyttää niiden tilalla haluttuja maapinnan tekstuureita.

Sijoittelin rakennusten ja muiden esineiden paikat valmiiksi Blenderissä ja sidoin ne kaikki yhteen emo-objektiin. Unityyn tuotuna tämä valmis paketti oli helppo sijoittaa kohdalleen emo-objektia siirtämällä, eikä lisäsäätöä tarvinnut tehdä.

Koska alueella on todella paljon puita, käytin Unityn automaattista puiden asettelutoimintoa. Koska toiminto lisäsi puita aivan kaikkialle, poistin manuaalisesti puut sieltä, minne ne eivät kuuluneet ja lisäsin muutamia puita käsin tärkeisiin paikkoihin. Alueelle tuli lopulta noin 25 000 puuta, mikä sellaisenaan tuhoaisi suorituskyvyn täysin.

Valaistuksen, varjojen, sumun ja muun lisäprosessoinnin asetukset ovat Creanexilla jo ennalta tehtynä lähes valmiina pakettina, joten työ niiden osalta oli vain lisätä tämä paketti skeneen. Valaistukseen täytyi tehdä hienosäätöä, mutta auringon suunnan, valon ja sumun sopivaksi säätämiseen ei kauaa mennyt.

8.2 Optimoinnit

Suurin parannus tulee ehdottomasti puiden instansoinnista. Käytämme Creanexilla Unityn GPUI-lisäosaa, ja se tuo merkittävän parannuksen suorituskykyyn. Koska GPUI käyttää omaa tapaansa käsitellä varjoja, tulee niistä hieman ylimääräistä säätötyötä hyvän lopputuloksen saamiseksi.

Toinen merkittävä optimointi on puiden LOD-tasotus. Lähellä oleville puille näytetään tarkin 3D-malli ja kauempana oleville toisiaan pienempilaatuinen malli, kunnes päädytään 2D-tasoon kaikkein kauimmaisille puille.

Viimeinen tässä skenessä oleva merkittävä optimointi on frustrum culling, joka piilottaa kameran näköalueen ulkopuolella olevat esineet, jolloin niitä ei tarvitse prosessoida. Frustrum cullingin lisäksi voisi käyttää occlusion cullingia, mikä piilottaisi myös kameran näköalueen sisässä olevat asiat, jotka ovat näkymättömissä jonkin takana. Occlusion culling ei ole tällä kertaa käytössä, koska alueella on melko vähän riittävän suuria näköesteitä.

9 UNREAL ENGINE 5 SKENE

9.1 Skenen rakentaminen

Aloitin rakentamaan maailmaa samaan tapaan kuin Unityssä, eli maapohjasta. Ongelmaksi ilmeni heti maapohjan mallin kääntäminen UE5:n maaksi. Unityä varten on työkalu, jolla tämä onnistuu helposti, mutta UE5:ssä sellaista ei näytä olevan helposti saatavilla edes kaupassa. Ensiksi minun täytyi kääntää maapohjan malli korkeuskartaksi Blenderissä, ja vasta sen jälkeen pystyin luomaan maan Unreal Engineessä. Toinen vaihtoehto olisi ollut käyttää maapohjan mallia sellaisenaan, mutta se olisi ollut hyvä teksturoida ennen tuontia pelimoottoriin, koska perinteisille malleille ei voi käyttää UE5:n hyviä maan maalaustyökaluja.

Käytin UE5 skenessä suurempaa maapohjaa ilman toista heikompileatuista ympäristön mallia, koska halusin kokeilla automaattista materiaalia. Näitä on muutama saatavilla ilmaiseksi Internetistä. Otin kokeiluun Youtube-käyttäjä Unreal Sensein ”master materialin” sen laajojen ominaisuuksien vuoksi. Ikävä kyllä se ei ole käytettävissä kaupallisesti, mutta kokeilumielessä tässä henkilökohtaisessa projektissa sitä voi käyttää. UE5-versio tästä maastosta ei ole tulossa simulaattorikäyttöön, ja jos tulee, niin materiaali pitää silloin poistaa. Materiaalin käyttäminen oli lopulta helppoa, kun sain lisättyä sen projektiin oikein. Se oli vakiona säädetty tuottamaan ruohoinen maa ja kalliota, jonka tekstuurit vaihdoin Unityn skeneä vastaaviksi. Automaattisen materiaalin käyttäminen nopeutti työtä huomattavasti, kun kaikki jyrkänteet tulivat suoraan kallioksi, eikä niitä tarvinnut itse maalata. Materiaalin päälle pystyy maalaamaan myös käsin, jota hyödynsin laskettelualueen peittämiseen eri tekstuurilla. Lopputulos olisi voinut olla vieläkin parempi, jos maapohjan mallin resoluutio olisi ollut suurempi.

Rakennusten mallit ja materiaalit pystyi lisäämään skeneen samoin, kuin Unityssä. Ainut ero on objektien skaala, joka on UE5:ssa eri kuin Unityssä. UE5 käyttää myös hieman erilaista normaalitekstuurin muotoa, jonka vuoksi nämä tekstuurit piti kääntää sopiviksi materiaalien asetuksissa.

Metsän rakentaminen tapahtuu samaan tapaan, kuten Unityssä. Puut voi maalata itse tai pelimoottori voi generoida haluamasi määrän puita maailmaan. Aluksi generoin 40 000 puuta, joista poistin kaikki mitkä olivat väärissä paikoissa, tai muuten vaan turhia. UE5:n puiden asettelutoiminto on huomattavasti Unityä parempi, koska puiden tiheyden voi manuaalisesti asettaa. Tämä estää puiden päällekkäin asettamisen, jota Unityssä yleensä tapahtuu.

9.2 Haasteet

Unreal tarjosi useita haasteita. Ensimmäiseksi tuli vastaan maaston muuttaminen Nanite-muotoon. Projektin maasto on hyvin suuri, mikä aiheutti myös suuren vaatimuksen tietokoneen välimuistille Nanite-käännösprosessin ajaksi. Muistin loppuminen kesken aiheutti editorin kaatuilua. Ongelman ratkaistua tuli toinen murhe, nimittäin valmis Nanite-maa oli hyvin epätasainen. Tämän lisäksi maan valaistus meni yksinkertaisesti sanottuna rikki. Ongelmaan perehdyttyäni kävi selväksi, että valaistuksen outous johtui siitä, että Nanitesta huolimatta valaistus lasketaan alkuperäisen maaston muotojen mukaan ja sitten vain näytetään Nanite-mallin päällä. Mallin, jonka muoto oli jotakin aivan muuta kuin piti. Maaston koon ja Nanite-prosessoinnin hitauden vuoksi joudun jättämään tämän ominaisuuden tutkimisen lähes kokonaan pois.

Hankaluuksia aiheutti aluksi myös kolmen eri kameran saaminen käyttöön samaan aikaan. Ratkaisu löytyi lopulta UE5:n monipelitoiminnosta. Jokainen kamera asetettiin eri pelaajaksi ja näytettiin vierekkäin oikeassa järjestyksessä. Seuraavaksi havahduin, että Lumen ei ollutkaan enää käytössä. Selvitettyäni asiaa kävi ilmi, ettei Lumen toimi yli kahdella aktiivisella kameralla. Löysin kyllä keinon ohittaa tämän rajoituksen, mutta silloin suorituskyky oli niin huono, ettei sitä ollut kannattavaa testata ollenkaan.

9.3 Optimoinnit

UE5 hyödyntää automaattisesti molempia culling-tekniikoita, mitkä edellisessä Unity-osiossa mainittiin. Sen lisäksi Unreal Enginen Nanite on myös eräänlainen

culling-keino, joka jättää käsittelemättä kameralle näkymättömiä mallien osia. Nante säättää myös mallien tarkkuutta katseluetäisyyden mukaan toimien automaattisena LOD-tekniikkana.

Unreal Engine sisältää myös esiasetetut skaalausasetukset grafiikan laadulle. Lähimpänä Unity-simulaattoreiden grafiikan laatua on medium-asetus, mutta tämä medium on hieman huonompi kuin Unityssä. High olisi tasoltaan taas huomattavasti parempi.

Uusi projekti käyttää vakiona TSR-tyyppistä reunanpehmenystä. TSR kuitenkin heikentää suorituskykyä sen verran, ettei sitä ole kannattavaa käyttää huolimatta sen parhaasta laadusta verrattuna muihin vaihtoehtoihin. Sen sijaan valitsin suorituskykymittauksiin TAA-reunanpehmenyksen, koska sillä ei käytännössä ole vaikutusta suorituskykyyn.

Skenessä on käytössä UE5:n volumetric-pilvet, jotka yrittävät simuloida oikeita pilviä ja niiden koostumusta. Asetuksella on noin 5 FPS:n vaikutus suorituskykyyn, mutta mitä luultavimmin sitä saa vähennettyä säätämällä sen laatua. Jätin asetuksen päälle sen visuaalisen merkityksen vuoksi. Unity-skenessä on myös liikkuvat pilvet, mutta ne eivät ole simuloituja, kuten UE5:n ovat.

10 MITTAUKSET

10.1 Miten mittaukset tehdään

Molempiin pelimoottoreihin on luotu toisiaan vastaavat skenet. Molemmissa skeneissä on lentorata, jota pitkin kamera-alusta lentää mittauksen aikana. Tähän alustaan on sidottu lumikissan malli ja sen sisässä on tarvittava määrä kameroita, eli yksi tai kolme testiskenaarion mukaan.

Kolmen kameran testissä kaikki kamerat renderöidään samalle 4K, eli 3840x2160-resolutiveiselle näytölle vierekkäin, hieman simulaattoria matkien. Sivujen kamerat osoittavat 45 astetta pois keskilinjasta. Kaikilla kameroilla on vaakatasossa 45 asteen näkökenttä. Yhden kameran testissä kameran näkökenttä on 100 astetta. Mittauksiin käytetyssä tietokoneessa on RTX 2070 SUPER -näytönohjain. UE5-pelimoottorista on käytössä versio 5.3. Jokaista testiskenaariota varten käytetään valmiiksi buildattua, eli käännettyä ohjelmaa, ettei taustalla oleva pelimoottorin editori vaikuta tuloksiin.

Nanitea on vielä erikseen testattu tyhjässä skenessä suuritarkkuisella mallilla, jota on monistettu muutama kappale. Skenessä kamera kiertää malleja. Mittaus tehdään kahdelta kierrokselta Naniten ollessa päällä ja pois.

Suorituskykymittauksissa käytetään MSI Afterburner -sovellusta, joka sisältää hyvät lokitoiminnot tulosten tallentamiseen. Keskiarvoisen ruudunpäivitysnopeuden lisäksi ohjelma tallentaa myös erilaisia huippuarvoja, kuten alimpien ja korkeimpien nopeuksien arvot, joista voi arvioida kuinka tasainen kokemus näytöllä on.

10.2 Mitattavat skenaarit

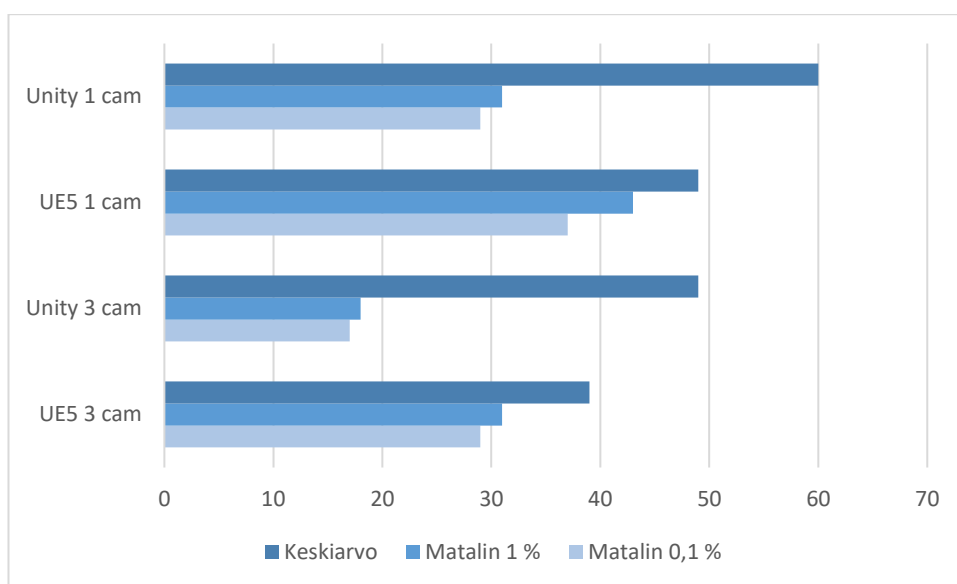
Molemmissa pelimoottoreissa mitataan skenen suorituskyky käyttämällä yhtä ja kolmea kameraa. Näistä tuloksista saadaan vertailtua ylimääräisten kameroiden vaikutus suorituskykyyn.

Unreal Engine 5:ssä suoritetaan enemmän mittauksia. Skenaarioihin kuuluu mitaukset yhdellä ja kolmella kameralla reunanpehmennys päällä ja pois, Lumen päällä ja pois, Nanite-maapohjalla ja ilman, sekä Unity-skenen laatua vastaavilla grafiikka-asetuksilla. Nanitea testataan myös omassa skenessä työkoneen mallilla.

11 TULOKSET

11.1 Unity vs Unreal Engine 5

Katsotaan aluksi, kuinka pelimoottorit vertautuvat toisiinsa. Kuviossa 1 on esitetty tulokset yhden ja kolmen kameran skeneistä, kun UE5:ssä on käytössä Unityä vastaavat kuvanlaatuasetukset.



KUVIO 1. Tulokset Unityn ja Unreal Engine 5:n suorituskyvystä yhdellä ja kolmella kameralla.

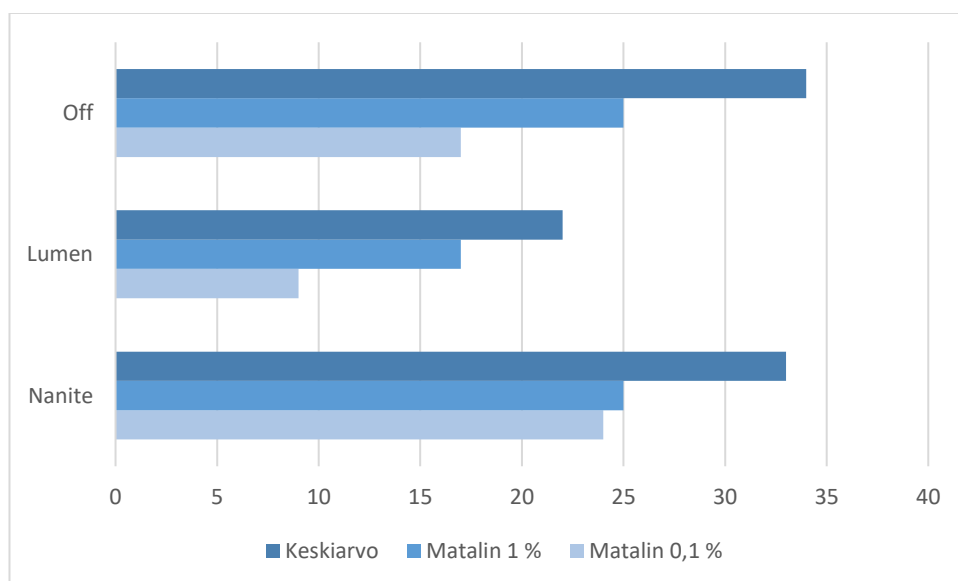
Kaaviossa on kuvattu testien keskiarvoinen ruudunpäivitysnopeus, mutta tämän lisäksi myös keskiarvot matalimpien prosenttien ja prosenttien kymmenyksen osalta. Keskiarvo kuvaa yleistä FPS-tasoa ja matalimmat arvot suorituskyvyn tasaisuutta. Mitä kauempana matalimmat arvot ovat keskiarvosta, sitä ikävämpi on käyttäjän kokemus.

Yhden kameran tuloksia verrattaessa voidaan todeta Unityn olevan hieman Unrealia nopeampi, mutta samalla on otettava huomioon, ettei Unrealin skeneä ole optimoitu samalle tasolle Unityn kanssa. Sanoisin, että pelimoottorit vastaavat toisiaan suorituskyvyssä. Molemmilla pelimoottoreilla kolme kameraa tuottaa noin 20 % pienemmän ruudunpäivitysnopeuden yhteen kameraan verrattuna.

Suorituskyvyn tasaisuudesta voi tehdä selkeän havainnon. Unityn matalimmat FPS-arvot ovat melkoisesti Unrealia pienemmät, joka tarkoittaa käyttäjälle enemmän epämiellyttäviä töksähdyksiä. Merkittävää on, että UE5:n kolmen kameran matalimmat suorituskykyarvot ovat tasoissa Unityn yhden kameran tuloksiin. Unityn suuremmasta keskiarvosta huolimatta nämä kokemukset tuntuvat käyttäjän näkökulmasta hyvin samanlaisilta.

11.2 Lumen ja Nanite

Unreal Engine 5:n eri ominaisuuksia mitatessa olen käyttänyt pelimoottorin oletus ”Epic”-skaalausasetuksia. Asetusten tuottama FPS on simulaattori- ja pelikäyttöön liian pieni käyttämälläni näyttönohjaimella. Tästä huolimatta seuraavia tuloksia voi verrata, koska FPS-erot skaalautuvat melko lineaarisesti saatavilla olevan näyttönohjaimen tehon mukaan. Kuviossa 2 on kuvattu tulokset ilman Lumenia ja Nanitea, sekä yksin Lumen ja Nanite käytössä.



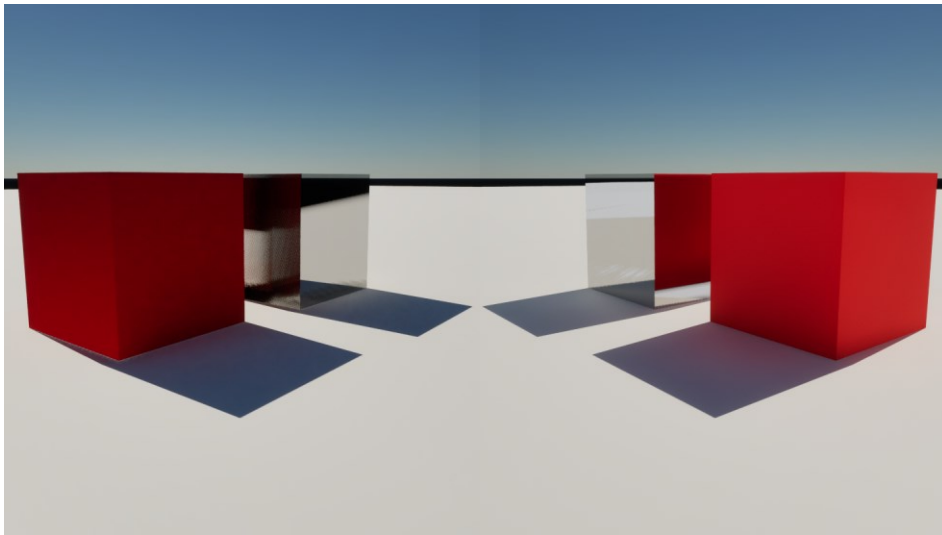
KUVIO 2. Lumen ja Nanite -toimintojen vaikutus suorituskykyyn.

Tuloksista voidaan todeta, että Lumen vie Epic-asetuksilla noin kolmanneksen näyttönohjaimen tehoista. Tämä on todella paljon vain yhdeltä toiminnolta, mutta vaikutus skenen visuaaliseen ilmeeseen on myös suuri. Kuvassa 3 on havainnollistettu Lumenin vaikutus valaistukseen.



KUVA 3. Perinteinen valaistus vasemmassa reunassa, Lumen oikealla.

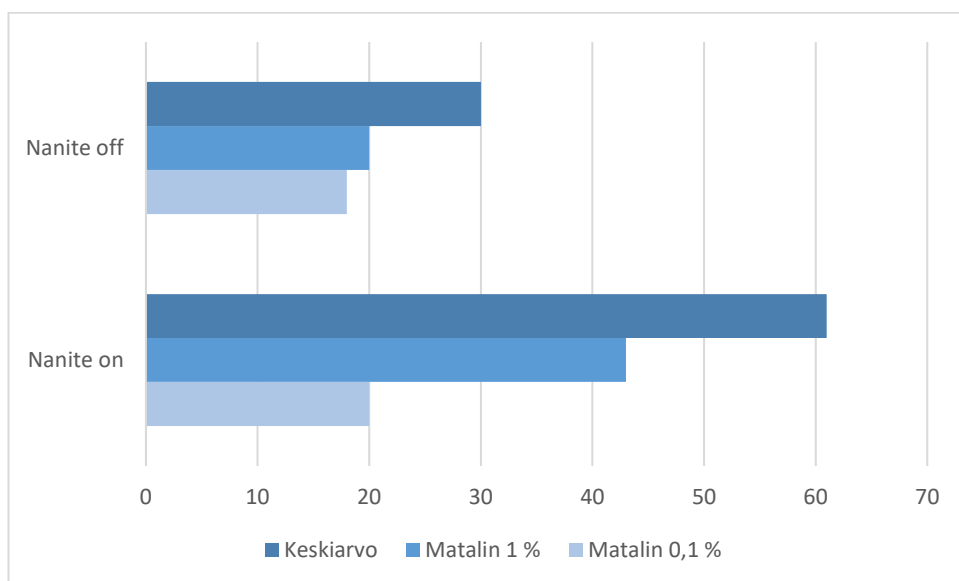
Lumenin vaikutus valon ja varjojen pehmeyteen, sekä epäsuoraan valaistukseen on merkittävä. Kuvasta 4 voidaan tämän lisäksi nähdä, että Lumen kykenee näyttämään heijastuksia sellaisista pinnoista, jotka eivät ole näytöllä. Tästä esimerkkinä on heijastus punaisen kuution takapinnasta peilikuutiossa. Myös heijastuneen valon väri siirtyy luonnollisesti esineen ympärille, jonka voi helposti nähdä varjossa. Perinteisessä valaistuksessa tätä efektiä on yritetty jäljitellä, mutta se ei ole yhtä näyttävä.



KUVA 4. Esimerkki Lumenin säteenseurannasta. Perinteinen valaistus vasemalla ja Lumen oikealla.

Skene-testien jälkeen Nanitesta on oikeastaan mahdotonta sanoa, miten se vaikuttaa suorituskykyyn maaston osalta. Tuloksissa Nanite oli keskiarvoisesti vain muutaman prosentin hitaampi asetus päällä. Kun tarkastellaan alimpia FPS arvoja, Nanite auttoi selvästi pitämään ruudunpäivitysnopeuden tasaisena.

Koska Naniten hyödyt eivät oikein näyttäytyneet projektin maastossa, toteutin ylimääräisen testin, jossa on muutama erittäin tarkka CAD-malli. Testiskenessä ei ole mallien ja Lumenin lisäksi mitään muuta suorituskykyä rasittavaa. Kuviossa 3 on kuvattuna tulokset testiskenestä.

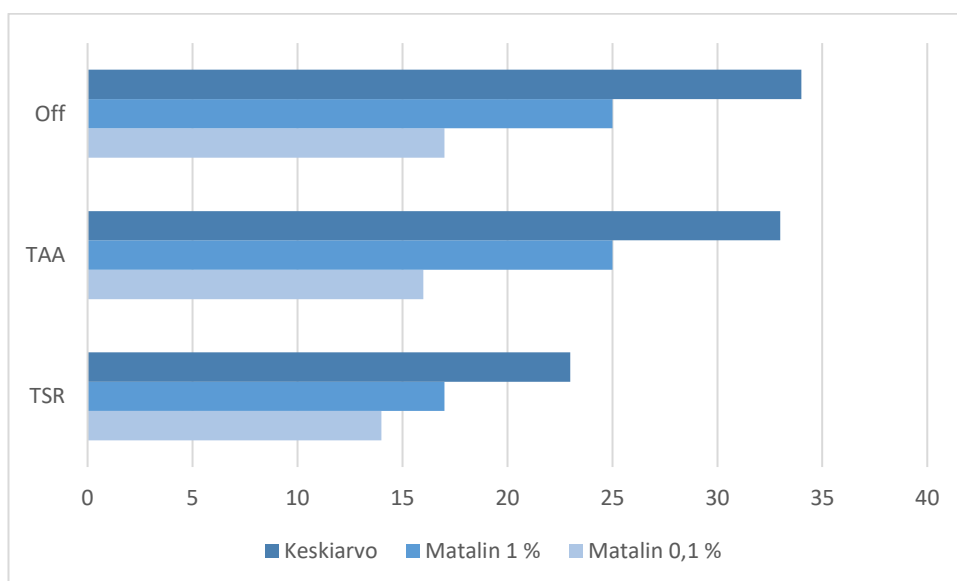


KUVIO 3. Naniten suorituskyky erittäin raskaassa skenessä.

Skenen yhteenlasketut noin 30 miljoonaa kolmiota toivat näyttöohjaimen polvilleen. Ilman Nanitea skenen keskiarvoinen FPS oli vain puolet siitä, mitä asetus päällä saavutettiin.

11.3 UE5 reunanpehmennys

Unreal Engine 5:ssä on valittavissa kolme erilaista reunanpehmennystekniikkaa, TAA, TSR ja FXAA. FXAA ei ole listattu alla olevaan kaavioon, koska jos sitä ei ole aivan pakko käyttää, niin sitä ei mielellään käytetä. Sen jättämä samaa kuva ei ole miellyttävän näköinen erityisesti pienemmän resoluution näytöillä. FXAA on suorituskyvyltään lähes ilmainen reunanpehmennystekniikka ja täten se olisi tasoissa TAA:n kanssa. Kuviossa 4 on nähtävillä tulokset reunanpehmennystekniikoiden suorituskyvystä.



KUVIO 4. UE5:n reunanpehmennystekniikoiden suorituskyvyn vertailu

Kuvanlaatunsa puolesta TSR on ylivoimaisesti paras, erityisesti silloin, kun renderöitävä resoluutio on alle näytön maksimiresoluution. Skaalausta käytettäessä TSR tulisi muita tekniikoita lähemmäs suorituskyvyssä ja kuvanlaadussa. TAA tarjoaa huomattavasti paremman kuvan samalla suorituskyvyn kustannuksella, kuin FXAA.

11.4 Onko pelimoottorin vaihto hyvä idea?

Tulosten perusteella UE5 vaikuttaisi olevan Unityä parempi pelimoottori Creanexin simulaattoreihin. Unreal Engine 5:n edut grafiikassa ja suorituskyvyn tasaisuudessa ovat testeissä omaa luokkaansa. Se myös omaa erinomaiset maastonmuokkaustyökalut. Lumenia voisi korkeasta suorituskykyvaatimuksestaan huolimatta todennäköisesti hyödyntää tunnelikone-simulaattoreissa, joka varmasti parantaisi skenen ilmettä valon heijastuksien ansiosta.

Vaihto ei tapahtuisi täysin ilman varjopuolia. Maaston tuonti Unityyn on melkoisesti nopeampaa, kuin UE5:ssä. Ainakin kunnes tähän keksitään jokin työkalu muuntamaan maaston malli korkeusdataksi. Unrealia ei myöskään voi käyttää vanhemmilla tietokoneilla korkean suorituskyvyn tarpeensa vuoksi. Lisäksi on useita muita simulaattorin tarvitsemia tutkimattomia ominaisuuksia, jotka voivat aiheuttaa ongelmia.

12 POHDINTA

Projekti oli mielenkiintoinen toteuttaa, sillä opin sen aikana paljon pelimootto-reista ja erityisesti UE5:n käytöstä. Hiihtokeskuksen maaston koon ja tarvittun työn vuoksi minulla ei ollut resursseja testata pelimootto-reita erilaisessa maas-tossa, joka olisi voinut tuottaa hyvinkin erilaiset tulokset. Pienellä otannalla ei saa kattavia tuloksia, mutta ainakin ne ovat suuntaa antavia. Olen silti tyytyväinen tuloksiin, koska mittauksista selvisi eroja, joita pelkkä perinteinen FPS-mittari ei kerro.

Maaston ja assettien luontiin meni reilusti enemmän aikaa, kuin osasin odottaa. Assettien lukumäärähän on pieni, mutta joistakin alueista oli hyvin vaikea löytää hyvää referenssimateriaalia. Maaston reuna-alueiden mallin löytämiseen ja toi-mintaan saamiseen meni myös huomattavasti aikaa. Toisaalta nyt, kun selvitys-työ on tehty, se sujuisi paljon nopeammin. Laakson pohjalla kulkee oikeasti tie ja siellä on monia rakennuksia, mutta jätin ne pois aikaa säästääkseni. Eipä simu-laattorissa sinne olisikaan asiaa.

En ollut ennen käyttänyt automaattista maaston materiaalia. Nyt sitä Unrealissa käytettyäni olen sitä mieltä, että sellainen täytyy jatkossa olla. Maastot tuntuvat kasvavan kokoa uusissa projekteissa, joten teksturoinnin automatisointi nopeut-taisi työtä huomattavasti ja jättäisi aikaa muulle viimeistelylle. Unrealin kaupassa on saatavilla valmiita automaattimateriaaleja, mutta olisi myös mielenkiintoinen kokemus luoda yksinkertaisempi versio itse.

Ehdin projektin aikana raapaisemaan UE5:n RVT:tä, eli lyhyesti virtuaalisia teks-tuureita, joilla voi vaikka sekoittaa maan ja kiven väliin jäävää rajaa luonnolliseksi ja huomaamattomaksi. RVT:llä voisi tehdä vaikka mitä muutakin, mutta joka ta-pauksessa sen ominaisuuksiin on jatkossa tutustuttava tarkemmin.

LÄHTEET

Balasubramanian, K. 2022. Game Engines: All You Need to Know. Gameopedia. Verkkosivu. Viitattu 27.3.2024. <https://www.gameopedia.com/game-engines-all-you-need-to-know-about/>

Buttfield-Addison, M. 2023. A comprehensive guide to the Unity Game Engine: Answers to Your Questions. Verkkosivu. Game Developer. Viitattu 2.4.2024. <https://www.gamedeveloper.com/production/a-comprehensive-guide-to-the-unity-2d-3d-game-engine>

Buttle, P. 2020. The Power Behind Video Games: A Look at Game Engines. Medium. Verkkosivu. Viitattu 27.3.2024. <https://medium.com/wetheplayers/the-power-behind-video-games-a-look-at-game-engines-2731315086e0>

Carpenter, N. 2023 a. Game devs say Unity's big change puts studios at risk. Polygon. Verkkosivu. Viitattu 21.4.2024. <https://www.polygon.com/23870247/unity-engine-pricing-model-install-fee>

Carpenter, N. 2023 b. Unity walks back some of its widely criticized pricing updates. Polygon. Verkkosivu. Viitattu 21.4.2024. <https://www.polygon.com/23885373/unity-technologies-install-fee-pricing-change>

Cowan, B. & Kapralos, B. 2014. A Survey of Frameworks and Game Engines for Serious Game Development. Luento. International Conference on Advanced Learning Technologies 7.-10.7.2014. Ateena.

Davis, N. & Burrows, D. 2022. 'Screens are the new horsepower': Leading luxury auto brand innovates with Unity. Verkkosivu. Viitattu 2.4.2024. <https://blog.unity.com/industry/screens-are-the-new-horsepower-leading-luxury-auto-brand-innovates-with-unity>

Good, O. 2020. How Lucasfilm used Unreal Engine to make The Mandalorian. Polygon. Verkkosivu. Viitattu 21.4.2024. <https://www.polygon.com/tv/2020/2/20/21146152/the-mandalorian-making-of-video-unreal-engine-projection-screen>

Gregory, J. 2018. Game Engine Architecture. Kolmas painos. E-kirja. CRC Press. Vaatii käyttöoikeuden. <https://learning.oreilly.com/library/view/game-engine-architecture/9781351974271/>

Unity Technologies. 2024 a. Plans and pricing. Verkkosivu. Viitattu 21.4.2024. <https://unity.com/products/unity-personal>

Unity Technologies. 2024 b. Unity Engine. Verkkosivu. Viitattu 2.4.2024. <https://unity.com/products/unity-engine>

Unity Technologies. 2024 c. Unity Personal. Verkkosivu. Viitattu 21.4.2024. <https://unity.com/products/unity-personal>

Unreal Engine. 2024. Licensing. Verkkosivu. Viitattu 21.4.2024. <https://www.unrealengine.com/en-US/license>

Unreal Engine. n.d. a. Bridge by Quixel. Verkkosivu. Viitattu 21.4.2024. <https://www.unrealengine.com/en-US/bridge>

Unreal Engine. n.d. b. Rivian's R1T features state-of-the-art HMI, designed, developed, and deployed in Unreal Engine. Verkkosivu. Viitattu 21.4.2024. <https://www.unrealengine.com/en-US/spotlights/rivian-s-r1t-features-state-of-the-art-hmi-designed-developed-and-deployed-in-unreal-engine>

Venter, H. & Ogterop, W. 2022. Unreal engine 5 character creation, animation and cinematics. E-kirja. Birmingham: Packt Publishing. Viitattu 21.4.2024. Vaatii käyttöoikeuden. <https://learning.oreilly.com/library/view/unreal-engine-5/9781801812443/>

Wright, D. & Narkowicz, K. n.d. Unreal Engine 5 goes all-in on dynamic global illumination with Lumen. Unreal Engine. Verkkosivu. Viitattu 21.4.2024. <https://www.unrealengine.com/en-US/tech-blog/unreal-engine-5-goes-all-in-on->