



Kaupunkimalli Unreal Engine - pelimoottoriin

Case VirtuaaliViipuri

Veetu Varala

OPINNÄYTETYÖ
Toukokuu 2022

Rakennusarkkitehdin tutkinto-ohjelma

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Rakennusarkkitehdin tutkinto-ohjelma

VARALA, VEETU:
Kaupunkimalli Unreal Engine -pelimoottoriin
Case VirtuaaliViipuri

Opinnäytetyö 33 sivua
Toukokuu 2022

Opinnäytetyön tavoitteena oli tutkia VirtuaaliViipuri-projektin mallinnuksien yhdistämistä yhtenäiseksi kaupunkimalliksi käyttäen alustana Unreal Engine -pelimoottoria.

Työ toteutettiin projektiluonteisesti ja sen pohjatuksena tutkittiin olemassa olevia pelimoottorialustaisia kaupunkimalleja. VirtuaaliViipuri-projektin menetelmät ja mallinnukset analysoitiin sekä tuotiin Unreal Engine -pelimoottoriin. Pelimoottorissa mallinnukset yhdistettiin kaupunkimalliksi ja optimoitiin. Työvaiheet ja siihen liittyvät ongelmat sekä ratkaisut dokumentoitiin.

Työn tuloksena toteutui yhtenäinen ja vapaasti tarkasteltava kaupunkimalli historiallisesta Viipurista. Unreal Engine -pelimoottorin todettiin olevan tehokas työkalu ja joustava alusta kaupunkimallien sovelluksille. Työn tuloksia voidaan hyödyntää niin VirtuaaliViipuri-projektin jatkokehityksessä kuin yleisesti kaupunkimallien kehitystyössä.

Asiasanat: kaupunkimalli, pelimoottori, Unreal Engine, VirtuaaliViipuri

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Construction Architecture

VARALA, VEETU:
City Model in Unreal Engine
Case VirtualViipuri

Bachelor's thesis 33 pages
May 2022

The objective of this thesis was to study and produce a city model in Unreal Engine from the materials of the VirtualViipuri project.

The study was carried out as a project. The project was primed with the study of the current game engine applicated city models. The methods and the building models from the VirtualViipuri project were analyzed and brought to Unreal Engine. Optimization and problem-solving were carried out in the Unreal Engine editor. The process and the solutions that were found are documented in the thesis.

As a result of the case work, a coherent and freely explorable city model of the historical Vyborg was produced. The Unreal Engine platform was found to be a flexible and powerful tool for city model applications. The results can be used in the further development of the VirtualViipuri project and might also prove to be useful in the general development work of city models.

Key words: city model, game engine, Unreal Engine, VirtualViipuri

SISÄLLYS

1	JOHDANTO	6
2	KAUPUNKIMALLI	8
2.1	Luokittelu ja kategoriat	8
2.2	CityGML ja tarkkuustasot	10
3	PELIMOOTTORI	11
3.1	Unreal Engine	11
3.1.1	Lisenssi	11
3.1.2	Tärkeitä ominaisuuksia	12
3.1.3	Ilmaiset materiaalit	14
3.1.4	Käyttäjäystävällisyys	14
4	PELIMOOTTORI KAUPUNKIMALLIN ALUSTANA	16
5	CASE VIRTUAALIVIIPURI	21
5.1	Lähtökohdat	22
5.2	Työvaiheet	23
5.2.1	Aineiston yhdistäminen	24
5.2.2	Mallinnusten optimointi	24
5.3	Ongelmat ja ratkaisut	27
6	POHDINTA	29
	LÄHTEET	31

ERITYISSANASTO

BIM	Building Information Model, rakennuksen tietomalli.
CAD	Computer Aided Design, tietokoneavusteinen suunnittelu.
Fotogrammetria	Kolmiulotteisten mittausten luominen valokuvien avulla.
Georeferointi	Kartan kiinnitys koordinaatistoon ja paikkaan.
GIS	Geographic Information System, paikkatietojärjestelmä.
Hahmontaminen	Tietokonegrafiikan kuvalaskentaprosessi.
Laserkeilaus	Lasersäteitä hyödyntävä mittausmenetelmä, jolla tuotetaan pistepilviä.
LOD	Level of Detail, geometrian tarkkuustaso.
Pistepilvi	Laserkeilauksessa tuotettu pisteistä koostuva kolmiulotteinen mittaustieto.
Polygoni	Tietokonegrafiikassa geometrian määrittävä monikulmio.
Tekstuuri	Geometrian pinnalla oleva 2D-kuviointi.
Verteksi	Tietokonegrafiikassa piste kolmiulotteisessa avaruudessa.

1 JOHDANTO

Kaupunkimallien kehitykseen ovat viime vuosina panostaneet useat eri kaupungit ja yritykset niin Suomessa kuin ulkomailla. Kaupunkimallien käyttöalustat ja -tarkoitukset vaihtelevat laajasti. Aiemmin kaupunkimalleja on hyödynnetty lähinnä suunnittelun apuna, mutta teknologisten edistysaskelten ansioista malleja voidaan nyt luoda ja esittää korkealla visuaalisella tarkkuudella. Näyttävät ja reaaliaikaisesti tarkasteltavat virtuaaliympäristöt ovat peliteollisuudessa olleet jo pitkään tavallisia, mutta nämä ympäristöt ovat harvoin kytköksissä todelliseen maailmaan. Fotogrammetrian ja pistepilvien avulla voidaan kuitenkin tallentaa digitaaliseen muotoon lähes täydellisiä kopioita todellisen maailman ympäristöistä.

Korkealla tarkkuudella on kuitenkin teknisiä haasteita. Aineiston käsittely vaatii siihen kykenevät ohjelmistot ja tarpeellisen suorituskyvyn laitteistolta. Esittelyalustan tulee olla saavutettava ja sen tulee tarjota osallistavuutta sekä visuaalista edistyskykyä. Ratkaisuna on aineistojen optimointi, johon peliteollisuudessa on panostettu jo pitkään. Peliteollisuudessa on tavoiteltu visuaalisesti korkeatasoista ja reaaliaikaista grafiikkaa, kun taas rakennusallalla kaupunkimalleja on kehitetty sisältämään suunnittelua avustavaa tietoa, kuten paikkatietoa. Näiden kahden alan lähestymistavat ovat kuitenkin kohtaamassa toisensa. Syitä ovat muun muassa virtuaaliympäristöihin kohdistuva kasvava kiinnostus ja rakennusalan työmenetelmien muuttuminen tietomallipainotteisimmiksi.

Rakennus- ja pelialan työmenetelmien, ohjelmistojen ja tiedostomuotojen yhteensopivuuden kehittäminen on ajankohtaista. Tämän opinnäytetyön tavoitteena on tutkia viimeisimmän peliteknologian mahdollisuuksia ja yhteensopivuutta BIM-mallinnustyökalulla tuotettujen korkean tarkkuustason rakennusmallien kanssa. Tähän opinnäytetyöhön valittiin työkaluksi ja kaupunkimallin alustaksi Unreal Engine -pelimoottori, joka edustaa viimeisintä peliteknologiaa. Opinnäytetyön päätavoitteena on tutkia yhtenäisen kaupunkimallin luontia hyödyntäen VirtuaaliViipuri -projektin mallinnuksia.

VirtuaaliViipuri-projektin lopulliset tavoitteet ovat historiallisen Viipurin visuaalisesti tarkka esittäminen sekä vapaasti liikuttava ja tarkasteltava virtuaaliympäristö (VirtuaaliViipuri 2022). Projektin mallinnuksia on tuotettu käyttäen tietomallinnukseen erikoistuneita ohjelmistoja ja työmenetelmiä, mutta projektin tavoitteet eivät liity niinkään semanttisen vaan visuaalisen kaupunkimallin luomiseen. VirtuaaliViipuri on edellä mainituista syistä hyvä esimerkkitapaus tutkia rakennus- ja pelialan työmenetelmien yhteensopivuutta.

Opinnäytetyön tavoitteena on myös edistää VirtuaaliViipuri-projektin alkuperäisiä tavoitteita eli virtuaalisesti liikuttavan Viipurin toteuttamista. Opinnäytetyössä ratkaistaan VirtuaaliViipuri-projektissa käytetyistä työmenetelmistä johtuvia ongelmia, mutta opinnäytetyön tuloksia ja sisältöä voidaan kokonaisuudessaan soveltaa myös muissa projekteissa sekä yleisesti kaupunkimallien kehitystyössä.

2 KAUPUNKIMALLI

Kaupunkimallilla tarkoitetaan kaupunkia tai sen osaa kuvaavaa kolmiulotteista esitystä eli 3D-mallia. Kaupunkimalli sisältää yleensä ainakin maaston ja rakennukset, mutta voi sisältää myös esimerkiksi puita ja liikenneverkostoja sekä paikkatietoa. (Julin ym. 2018)

Kaupunkimalleja hyödynnetään usein suunnittelun apuna, sillä ne kuvaavat todellisuuden selkeällä ja helposti hahmotettavalla tavalla. Kaupunkimalleissa voidaan hyödyntää visuaalisen esityksen lisäksi myös visuaalisesti havainnollistamatonta materiaalia. Tämä materiaali voi tarkoittaa muun muassa historiallista ja akustista tietoa tai muuta статистиikkaa, jota voidaan sisällyttää kaupunkimallin geometriseen materiaaliin kuten rakennuksiin. (Billen ym. 2014)

Kaupunkimallien luomisprosessi vaihtelee laajasti. Tavanomainen kaupunkimalli johdetaan useiden eri tekniikoiden yhdistelmänä, johon kuuluvat esimerkiksi fotogrammetrinen kuvantaminen, laserkeilaus ja viivapiirrokset (Biljecki ym. 2015).

2.1 Luokittelu ja kategoriat

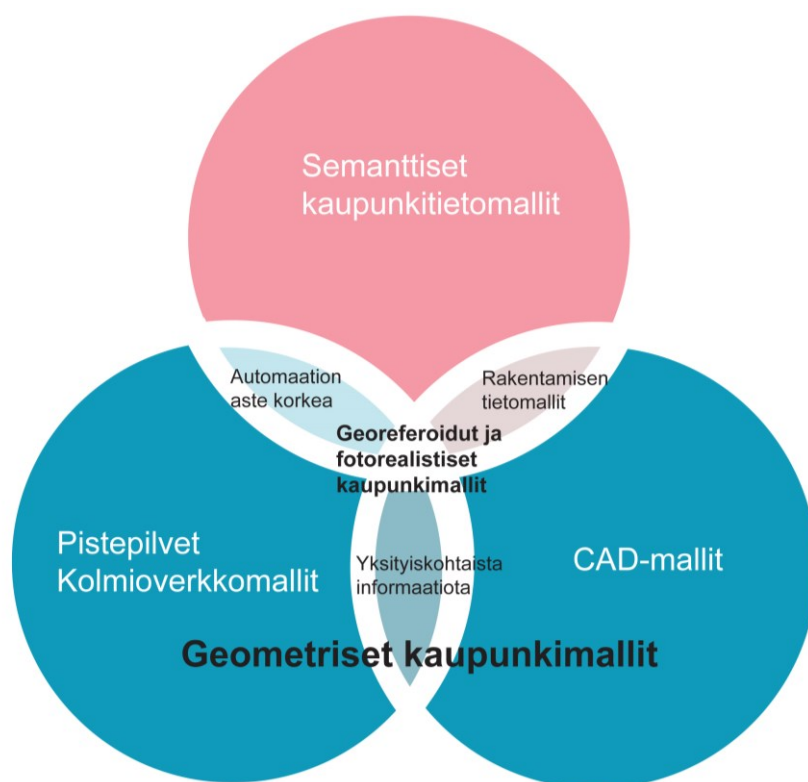
Kaupunkimallin määritelmä on monitulkintainen. Kaupunkimallia on terminä käytetty useissa erilaisissa tapauksissa liittyen kolmiulotteiseen rekonstruktioon ja semanttisiin malleihin. Useita erilaisia kaupunkimallialustoja on saatavilla, mikä myös hankaloittaa kaupunkimalleista käytävää keskustelua. (Julin ym. 2018) Määritelmän epäselvyydestä ja alustojen laajasta valikoimasta johtuen myös kaupunkimallien luokittelu on haastavaa. Luokittelutapoja kuitenkin löytyy esimerkiksi käyttötarkoitukseen ja soveltamistapaan liittyen.

Kolmiulotteisia kaupunkimalleja sovelletaan GIS/CAD-, virtuaalikartta- ja pelimoottorialustoilla. GIS/CAD-alustoihin pohjautuvat kaupunkimallit ovat suunnattu lähinnä ammattilaisten kuten kaupunkisuunnittelijoiden ja arkkitehtien käyttöön. (Julin ym. 2018)

Virtuaalikartta (virtual globe) kuten Google Earth on suunnattu yleiseen käyttöön, ja se on helposti saavutettavissa selaimen välityksellä. Virtuaalikartat sisältävät yleensä paikkatietoja ja niissä voi liikkua vapaasti vaihtaen tarkastelukulmaa. (Julin ym. 2018)

Pelimoottoreita hyödynnetään sovelluksissa, joissa tarvitaan esimerkiksi interaktiivisuutta ja korkeatasoista visuaalista realismia. Näitä sovelluksia käytetään yleensä arkkitehtuurin visualisointiin liittyvissä tehtävissä ja ne harvoin sisältävät GIS/CAD-ominaisuuksia tai ovat liittyneenä karttakoordinaatistoihin. (Julin ym. 2018)

Enni Airaksinen esittää diplomityössään kaupunkimallien luokitteluun kaksi eri pääkategoriaa: semanttiset ja geometriset kaupunkimallit. Geometrisillä kaupunkimalleilla tarkoitetaan pistepilviä, kolmioverkkomalleja ja CAD-malleja. Geometriset mallit voidaan jakaa vielä keskenään niiden automaatioasteen perusteella (KUVIO 1). Geometriset kaupunkimallit sisältävät geometriaa ja tekstuuria. Semanttiset kaupunkimallit sisältävät geometrian lisäksi myös tietoa. (Airaksinen 2017)



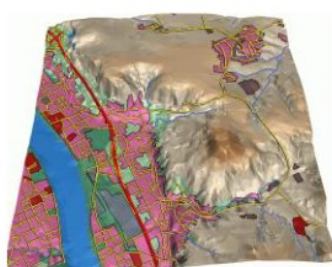
KUVIO 1. Kaupunkimallien kategoriat ja niiden limittyneisyys havainnollistettuna. (mukaillen Airaksinen, E. 2017)

2.2 CityGML ja tarkkuustasot

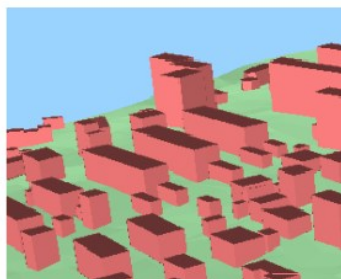
CityGML on avoin tietomalli, joka perustuu XML-formaattiin. CityGML on hyväksytty tietomallistandardi ja sen tavoitteena on kehittää kaupunkimalleihin liittyvää luokittelua sekä samalla kaupunkimallien ylläpidon tehokkuutta. (OGC 2012)

CityGML kuvaa mallinnukset viidellä eri tarkkuustasolla. Luokittelulla voidaan määritellä kaupunkimallin tarkkuustaso alustasta riippumatta.

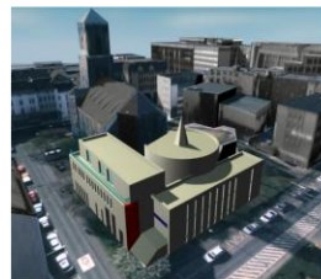
LOD0 on CityGML tarkkuustasoista yksinkertaisin ja sisältää korkeustietoa vain maastosta. LOD1 kuvaa maaston ja rakennukset laatikkomaisina massoina. Rakennusten kattomuotoa ei ole esitetty mallissa. LOD2 tuo lisää yksityiskohtia ja muotokieltä rakennuksiin sekä esittää karttaa jakavien elementtien, kuten teiden ja tonttien pintamateriaaleja. LOD3 on tarkka esitys rakennusten arkkitehtuurista. Tarkimmassa LOD4-tarkkuustasossa rakennuksissa esitetään myös sisätilat kalusteineen. (OGC 2012)



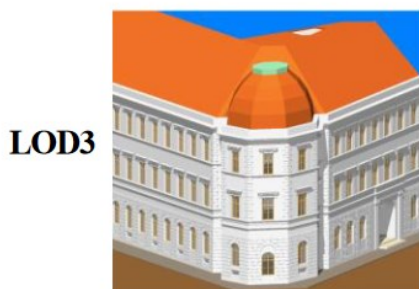
LOD0



LOD1



LOD2



LOD3



LOD4

KUVA 1. CityGML-standardin viisi eri tarkkuustasoa. (OGC 2012)

3 PELIMOOTTORI

Pelimoottori on digitaalinen alusta, joka tarjoaa esimerkiksi seuraavia toiminnallisuuksia: grafiikan hahmontaminen, tiedostojärjestelmä, hiirellä ja näppäimistöllä ohjaaminen, verkkotoiminnot sekä äänentoisto (Freiknecht ym. 2016). Pelimoottorit tekevät tietokonepelien tekemisestä helppoa yhdistelemällä useita erillisiä komponentteja yhdeksi kokonaisuudeksi (Barczak & Woźniak 2019). Pelimoottoreita voidaan joustavuuden, laajan työkaluvalikoiman ja toiminnallisuuden ansiosta hyödyntää lukuisissa eri sovelluksissa myös peliteollisuuden ulkopuolella.

3.1 Unreal Engine

Unreal Engine on Epic Games -yhtiön luoma pelimoottori. Pelimoottorin ensimmäinen versio on julkaistu Unreal-pelin yhteydessä vuonna 1998 (Barczak & Woźniak 2019). Pelimoottorin uusin versio on Unreal Engine 5, ja se julkaistiin vuoden 2022 keväällä (Unreal Engine 2022a). Unreal Engine tukee peleille ja sovelluksille seuraavia julkaisualustoja: Windows, Linux, Mac OS, iOS, Android, Playstation 4, Xbox One, HTML5 ja virtuaalitodellisuuslaitteita kuten SteamVR, HTC Vive ja Oculus Rift. Unreal Engine tukee C++ ohjelmointikieltä, ja sen ominaisuuksiin kuuluu visuaalisesti käytettävä Blueprint-ohjelmointijärjestelmä (Barczak & Woźniak 2019). Tässä opinnäytetyössä on pääasiassa tutkittu ja hyödynnetty Unreal Engine-pelimoottorin uusinta versiota.

3.1.1 Lisenssi

Unreal Engine on ilmaiseksi ladattava ohjelmisto. Ohjelmiston käyttö on ilmaista opetuskäytössä, sisäisissä projekteissa ja kaupallisissa projekteissa, joiden tulot jäävät alle asetetun rojaltykynnyksen. Epic Games perii 5 % rojalteina, kun Unreal Engine -pelimoottoria hyödyntävästä tuotteesta saadut bruttotulot ylittävät miljoona dollaria. (Unreal Engine 2022b)

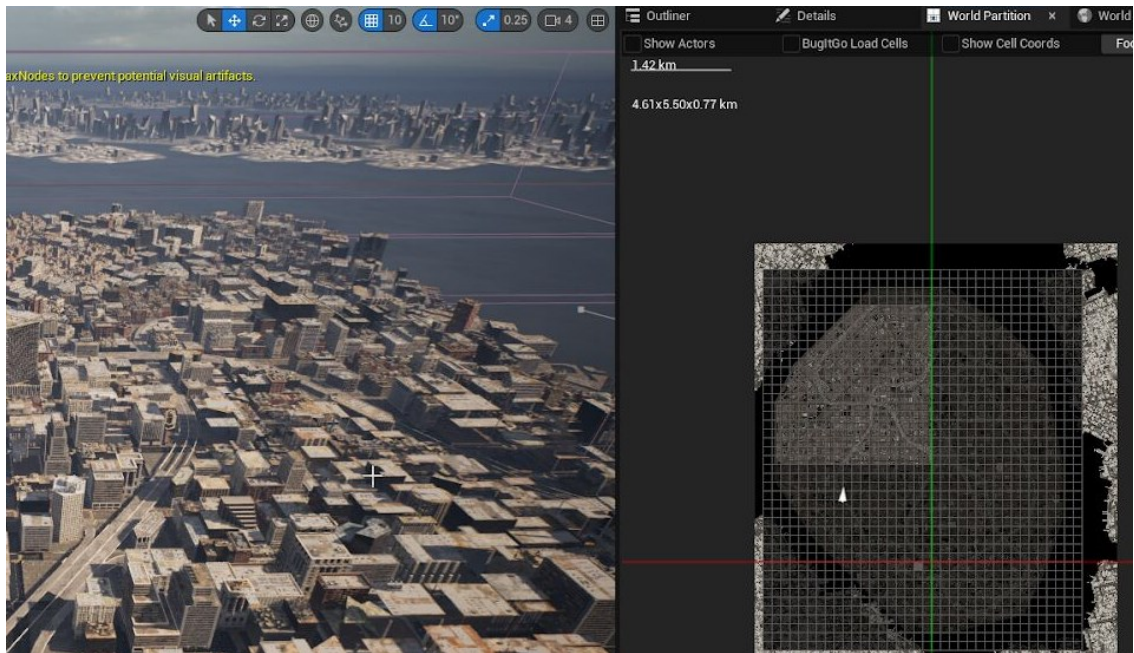
3.1.2 Tärkeät ominaisuudet

Tässä osiossa esitellään lyhyesti Unreal Engine -pelimoottorin teknisiä ominaisuuksia, joiden perusteella se valittiin tämän opinnäytetyön kaupunkimallin alustaksi. Luetellut ominaisuudet ovat merkittäviä kaupunkimallien ja muiden laajojen sekä joustavuutta vaativien projektien toteuttamisen kannalta.

Työmenetelmien yhteensopivuutta (pipeline integration) edistää muun muassa tuki useille eri tiedostomuodoille. Datasmith on kokoelma työkaluja ja lisäosia, joiden avulla taataan ongelmaton tiedostojen tuonti Unreal Engine-pelimoottoriin useista eri mallinnusohjelmista, mukaan lukien useista eri CAD- ja BIM- formaateista. Datasmith on suunniteltu erityisesti peliteollisuuden ulkopuolisille käyttäjille. (Unreal Engine 2022c)

Unreal Engine tukee tuotantovaiheessa myös Python-ohjelmointikielen komentosarjoja, ja pelimoottorilla on työkalu myös visuaaliselle tiedonkäsittelyn ohjelmoinnille (Visual Dataprep). Tiedonkäsittelyn ohjelmoinnilla voidaan luoda ”reseptejä” suorittamaan automaattisia toimintoja tuontimateriaaleille. Toiminnot voivat liittyä esimerkiksi tekstuurien ja materiaalien korvaamiseen, automaattiseen LOD-tasojen luomiseen tuotavasta geometriasta tai tarpeettoman geometrian tunnistamiseen ja poistamiseen tuonnin yhteydessä. (Unreal Engine 2022d) Unreal Engine tukee lisäosan avulla myös laserkeilausmenetelmillä tuotettuja pistepilviä. Pistepilviä on mahdollista visualisoida ja muokata suoraan pelimoottorissa. (Unreal Engine 2022e)

Virtuaaliympäristön luomista (world building) helpottaa muun muassa ympäristöä automaattisesti jaotteleva komponentti World Partition, jonka avulla laajoja ympäristöjä voidaan käsitellä pienempinä osa-alueina (KUVA 2). Tämä mahdollistaa esimerkiksi laajan ympäristön muokkaamisen samanaikaisesti useamman henkilön toimesta. (Unreal Engine 2022f)



KUVA 2. World Partition -ominaisuus käytössä Unreal Engine -pelimoottorin kaupunkiesimerkissä. (mukaillen Unreal Engine 2022f)

Virtuaalinen geometriajärjestelmä (Nanite) on Unreal Engine 5 -pelimoottorin ainutlaatuinen optimointitekniikka. Ominaisuus perustuu älykkääseen hahmontamistekniikkaan, joka päättelee ruudulle piirrettävän geometrian tarkkuustason. Nanite toimii siis automaattisena LOD-järjestelmänä ja mahdollistaa esimerkiksi käsittelemättömien fotogrammetrialla tuotettujen verkkomallien käytön reaaliaikaisissa sovelluksissa. (Unreal Engine 2022g)

TAULUKKO 1. Verkkomallin tiedostokoon vertailu Naniten ja tavallisen LOD-järjestelmän välillä. (mukaillen Unreal Engine 2022g).

Verkkomalli	Tavallinen	Nanite
Polygonit	1 545 338	1 545 338
Verteksit	793 330	793 330
LOD-tasot	4	-
Tiedostokoko (Mt)	148.95	19.64

Lumen on dynaaminen valaistus- ja heijastusjärjestelmä (KUVA 3), jonka ansiosta virtuaalisen ympäristön valaistusta voidaan muuttaa reaaliaikaisesti. Yleensä realistinen ja reaaliaikainen valaistus on vaatinut erillisten valaistustekstuureiden (lightmap) luomista kullekin valaistusasetukselle. (Unreal Engine 2022h)



KUVA 3. Ennen ja jälkeen Lumen-järjestelmän käyttöönoton. (mukaillen Unreal Engine 2022h)

3.1.3 Ilmaiset materiaalit

Työmenetelmien yhteensopivuutta ja teknistä suorituskykyä edistävien ominaisuuksien lisäksi Unreal Engine -pelimoottori tarjoaa käyttäjilleen paljon ilmaista materiaalia hyödynnettäväksi peleissä ja muissa sovelluksissa.

Quixel Bridge on Unreal Engine -pelimoottorin lisäosa, joka tarjoaa ilmaiseen käyttöön laajan kirjaston fotogrammetrisesti tuotettuja verkkomalleja (Megascans) ja tekstuureja. Megascans-tuotteita on hyödynnetty esimerkiksi Disney-elokuvissa niiden korkean visuaalisen tarkkuuden ja laadun ansiosta. Unreal Engine -kauppapaikka (Marketplace) tarjoaa myös tuhansia ilmaisia ja maksullisia lisäosia sekä muita materiaaleja. Kauppapaikalla on sekä Epic Games -yrityksen että Unreal Engine -käyttäjyhteisön luomaa materiaalia. (Unreal Engine 2022i)

3.1.4 Käyttäjäystävällisyys

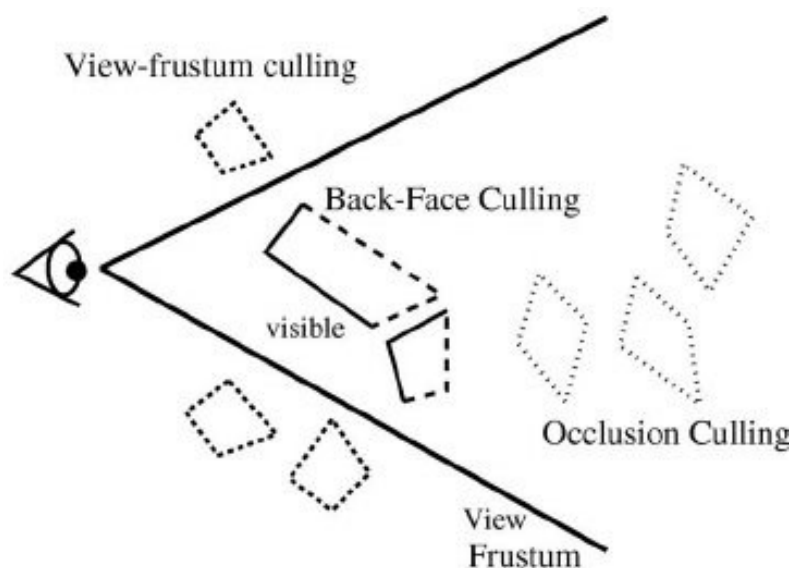
Unreal Engine -pelimoottori voi olla haastava alusta aloittelijoille verrattuna sen kilpailijaan Unity-pelimoottoriin. Lukuisten valikoiden ja asetusten määrä tekee Unreal Engine -pelimoottorin käyttöönottamisesta haasteellista. Toisaalta esi-

merkiksi sen Blueprint-työkalu tarjoaa mahdollisuuden visuaaliseen ohjelmointiin, jolloin tavanomaisia ohjelmointikielitäitoja ei tarvita. (Barczak & Woźniak 2019)

Unreal Engine -ohjemateriaali on avoin kaikille käyttäjille verkkosivuston muodossa. Ohjekirjasto on laaja ja sisältää selitykset kaikkiin pelimoottorin ominaisuuksiin. Lisäksi saatavilla on useita virallisia esimerkkiprojekteja, jotka esittelevät ominaisuuksia käytännön sovellusten kautta. Esimerkkejä voidaan käyttää alustoina toisille projekteille tai niiden osia voidaan liittää käyttäjän omiin sovelluksiin. Unreal Engine -pelimoottorista on sen ilmaisen lisenssin ansiosta saatavilla paljon myös kolmannen osapuolen opetusmateriaalia Unreal Engine -yhteisön kirjoituspaloilla ja verkkosivuilla kuten YouTube-videopalvelussa.

4 PELIMOOTTORI KAUPUNKIMALLIN ALUSTANA

Peliteollisuus on ollut pitkään edelläkävijä tietokonegrafiikan sovelluksissa. Vuorovaikutus virtuaaliympäristöjen kanssa, 3D-grafiikka ja sen optimointi ovat pääosin kehittyneet peliteollisuuden ansiosta. Optimointi lisää virtuaaliympäristöjen saavutettavuutta useilla eri laitteistoilla suorituskyvystä riippumatta. Optimointitapoja ovat esimerkiksi tekniikat, joissa katselualueen ulkopuolelle jäävää objekti ei piirretä (frustum culling) ja objektin piilottaminen, kun sen edessä on toinen objekti (occlusion culling) (KUVA 4). Peliteollisuus on kehittänyt eteenpäin myös sovellusten vuorovaikutteisuutta ja useiden käyttäjien järjestelmiä. (Virtanen ym. 2017)

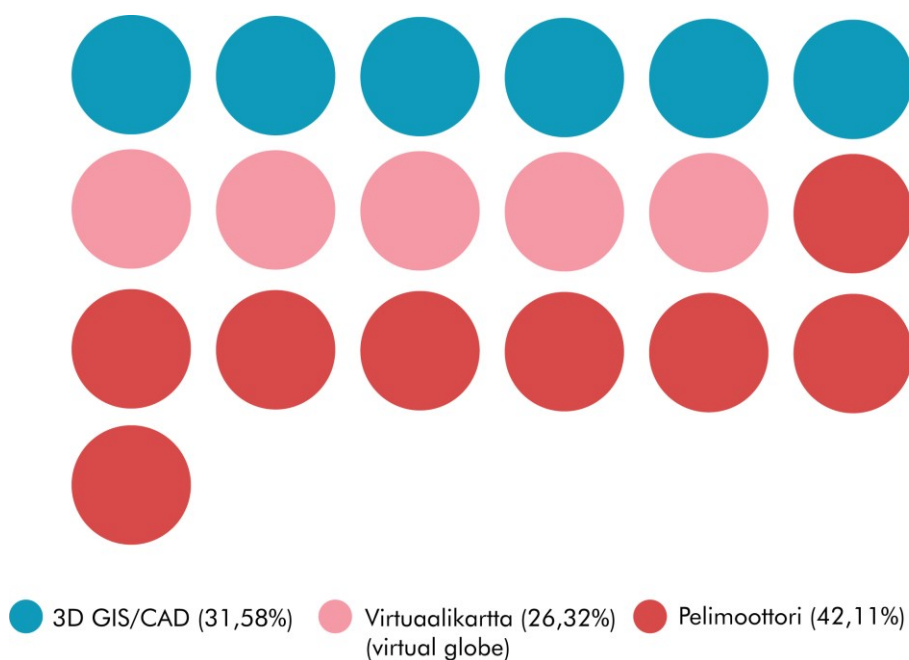


KUVA 4. Peliteollisuudessa yleisesti hyödynnettyjä optimointitekniikoita. (Cohen-Or ym. 2001)

Peliteollisuudessa kehitettyjä menetelmiä ja teknologioita voidaan hyödyntää kaupunkimalleissa ja niiden sovelluksissa. Aihe vaatii kuitenkin vielä lisää tutkimusta, jotta löydettäisiin uusia työmenetelmiä ja tapoja hyödyntää peliteollisuuden järjestelmiä kaupunkimallien kehitystyössä. (Virtanen ym. 2017) Pelimoottorit eivät vielä tarjoa suoraa yhteensopivuutta georeferoidun paikkatiedon tai karttaprojektioiden kanssa. Pelimoottoreita hyödynnetään kuitenkin suunnittelu- alan GIS/BIM-sovelluksissa sen tarjoaman reaaliaikaisen grafiikan ja visuaalisen edistyneisyyden ansiosta. (Julin ym. 2020)

Suosituimmat pelimoottorit kaupunkimallien sovelluksissa ovat Unity- ja Unreal Engine -pelimoottorit. Ne tarjoavat laajat käyttäjäyhteisöt ja ohjekirjastot, jotka helpottavat käyttöönoton aloittamisessa. Pelimoottoreiden kehitystyössä on aloitettu huomioimaan muitakin aloja kuin peliteollisuus. Lisääntyneet tiedonkäsittelytyökalut ja lisäosat tekevät pelimoottoreista kiinnostavia alustoja kaupunkimallien sovelluksiin. (Julin ym. 2020)

Julinin ym. vuoden 2018 tutkimuksen perusteella pelimoottori on suosittu kaupunkimallin alusta Suomessa (KUVIO 2). Kaupunkimalleille ei ole muodostunut yhtenäistä alustaa, joka kattaisi kaikki käyttötarkoitukset ja tekniset vaatimukset. Yhdellä kaupungilla saattaa olla kaupunkimalleja usealla eri alustalla. Näihin kaupunkimalleihin lukeutuu myös tutkimushankkeita kuten monialustainen VirtualOulu. (Julin ym. 2018)



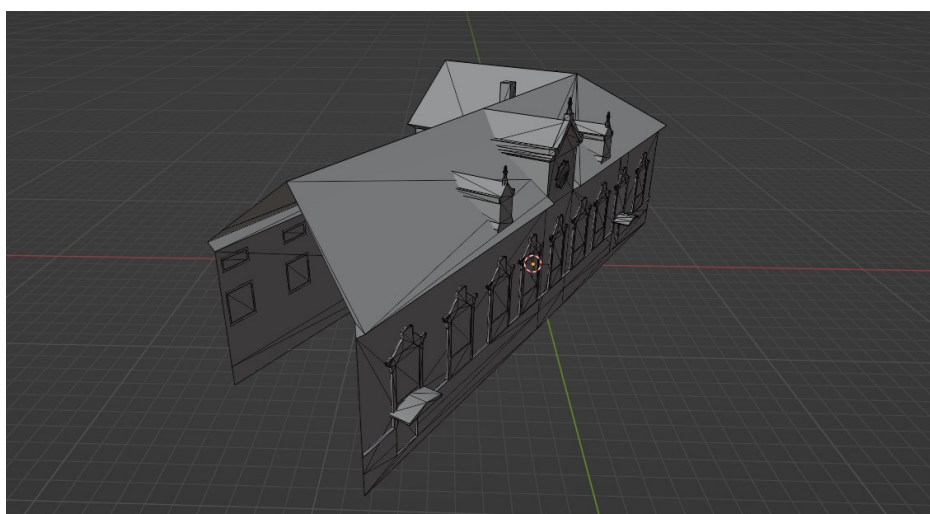
KUVIO 2. Tutkimuksen tulos 19 eri kaupunkimallin käyttämästä alustasta Suomessa. (mukaillen Julin ym. 2018)

VirtualOulu on Oulun yliopiston luoma tutkimus- ja kaupunkimallihanke, jonka tavoitteena on toteuttaa yhteistyöhön perustuva virtuaalinen kaupunki. Yhteistyöhön perustuva kaupunkimalli on avoimesti tarkasteltava ja muokattava muidenkin kuin alkuperäisten tekijöiden toimesta. Rakennuksia puretaan ja kaupunkiympäristössä tapahtuu jatkuvasti muutoksia, jolloin esimerkiksi arkkitehdit

voivat tuoda uudet rakennukset avoimeen kaupunkimalliin. VirtualOulu-hankkeen tavoitteena on olla ensimmäisestä persoonasta (first-person) tarkasteltava digitaalinen ”peilikaupunki”. Tämä tarkoittaa, että kaupunkiympäristö on tilallisesti todellisuutta vastaava ja esimerkiksi maaston muodot johdetaan laserkeilausaineistosta. VirtualOulun tavoitteena on myös kaupunkimallin saavutettavuus verkkoselaimessa. (Alatalo ym. 2016)

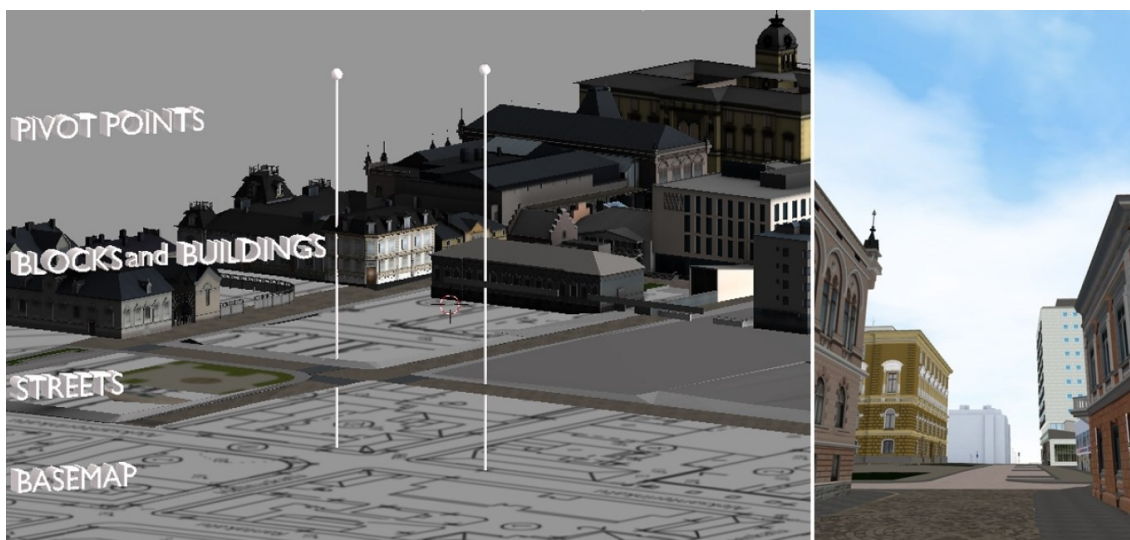
VirtualOulu-hankkeessa lähtöaineistoa varten rakennusten julkisivut valokuvattiin ja piirustukset muunnettiin digitaaliseen muotoon. Myös laserkeilausta käytettiin kolmiulotteisten pintojen arkistoinnissa. Rakennusten mallinnus tapahtui pääosin manuaalisesti ja työ jaettiin yhdeksän korttelin ryhmiin. Korttelien mallinnustyön toteutti kolme eri yritystä, sillä samalla haluttiin simuloida laajan kaupunkimallin luomista useiden tekijöiden yhteistyönä. Hankkeessa luotiin yhteiset työmenetelmät rakennusten mallintamista varten, jotta yksittäisten mallinnusten yhdistäminen laajaksi yhtenäiseksi kokonaisuudeksi onnistuisi ongelmitta. (Alatalo ym. 2016)

VirtualOulu-kaupunkimallin rakennukset ovat ladattavissa avoimena datana Git-versionhallintapalvelusta. Rakennukset on mallinnettu hyödyntäen Blender-mallinnusohjelmistoa eli mallinnusten lähdeaineistot ovat saatavilla kyseisen ohjelman tiedostomuodossa. (Alatalo ym. 2016) Tarkastelussa huomataan, että vain rakennusten ulkokuori on mallinnettu (KUVA 5). Rakennuksissa on myös käytetty vain yhtä tekstuuritiedostoa.



KUVA 5. VirtualOulu-hankkeen mallinnus ilmaisessa Blender-ohjelmistossa. (Varala 2022)

Kortteleiden yhdistäminen toteutettiin asettamalla jokaiselle korttelille oma kiintopisteensä koordinaatistossa. Korttelien omat kiintopisteet vastaavat kaupunkimalliin valmiiksi asetettuja kiintopisteitä (pivot points). Korttelit myös kierrettiin oikeaan asentoon jo mallinnusvaiheessa. Tämän ansioista korttelit voitiin asettaa suoraan oikeille paikoilleen yhdistetyssä kaupunkimallissa (KUVA 6). (Alatalo ym. 2016)



KUVA 6. VirtualOulu-kaupunkimallin yhdistämisessä hyödynnettiin pohjakarttaa. (Alatalo ym. 2016)

VirtualOulu-kaupunkimallia on hyödynnetty erilaisissa kaupunkimallin sovelluksissa kuten verkkoselaimessa tarkasteltavassa katutasen virtuaalikokemuksessa ja pelimoottorialustan pohjalta tuotetuissa pelisovelluksissa. Pelisovelluksia ovat esimerkiksi Street Art Gangs, joka on puhelimilla pelattava hybriditodellisuuspeli (mixed reality) sekä Historical Oulu, jossa historiallisia valokuvia kaupungista voidaan tarkastella niiden todellisessa ympäristössä kaupunkimallin sisällä. (Alatalo ym. 2016)

Usealla muullakin Suomen kaupungilla on pelimoottorialustainen kaupunkimalli. Nämä kaupunkimallit ovat usein geometrisia ja niiden käyttötarkoitus on visuaalinen. Esimerkiksi Tampereella on tutkittu Unity-pelimoottoria kaupunkimallin alustana, ja projekti on ladattavissa Tampereen kaupungin verkkosivustolta.

Tampereen Unity-kaupunkimalli on johdettu käyttäen apuna laaja lähdeaineistoa kuten avointa paikkatietoa, ilmakuvia, pistepilviä ja 3D-malleja rakennuksis-

ta. Kaupunkimalliin on sisällytetty myös kaavoitushankkeiden mallinnuksia. Aineisto voidaan tuoda Unity-projektin ulkopuolelle FBX-formaatissa, jolloin kaupunkimallia voidaan hyödyntää myös muissa pelimoottoreissa. Kaupunkimalli ei sovellu suunnittelutehtäviin, mutta sitä voidaan käyttää esimerkiksi visualisoinnissa. (Tampereen kaupunkimittaus 2018)

Tampereen Unity-kaupunkimallissa käytetyn aineiston projektioksi ilmoitetaan ETRS-GK24, joka on Tampereen alueen karttaprojektio (Tampereen kaupunkimittaus 2018). Projektio liittyy Suomen aluekohtaiseen EUREF-FIN -tasokoordinaattijärjestelmään (Tampereen kaupunki 2022). Aineiston perusteella Tampereen Unity-kaupunkimalli on georeferoitu, mutta käytännössä kaupunkimalli toimii pelimoottorin paikallisessa koordinaatistossa. Suoraa yhteyttä tai linkkiä paikkatietoon ei ole ilman muunnosta pelimoottorin ja todellisen karttaprojektion välillä.

Helsingin kaupungista on myös toteutettu pelimoottorialustainen kaupunkimalli, joka kattaa tiettyjä osia Helsingistä. Virtual Helsinki -projektissa on hyödynnetty muun muassa fotogrammetrisesti tuotettuja verkkomalleja ja Unreal Engine -pelimoottoria. Virtual Helsinki -projektin lopputuotteena on toteutettu VR-laseilla tarkasteltavat 360-animaatiot kaupunkimallista (KUVA 7). (Zoan 2022)



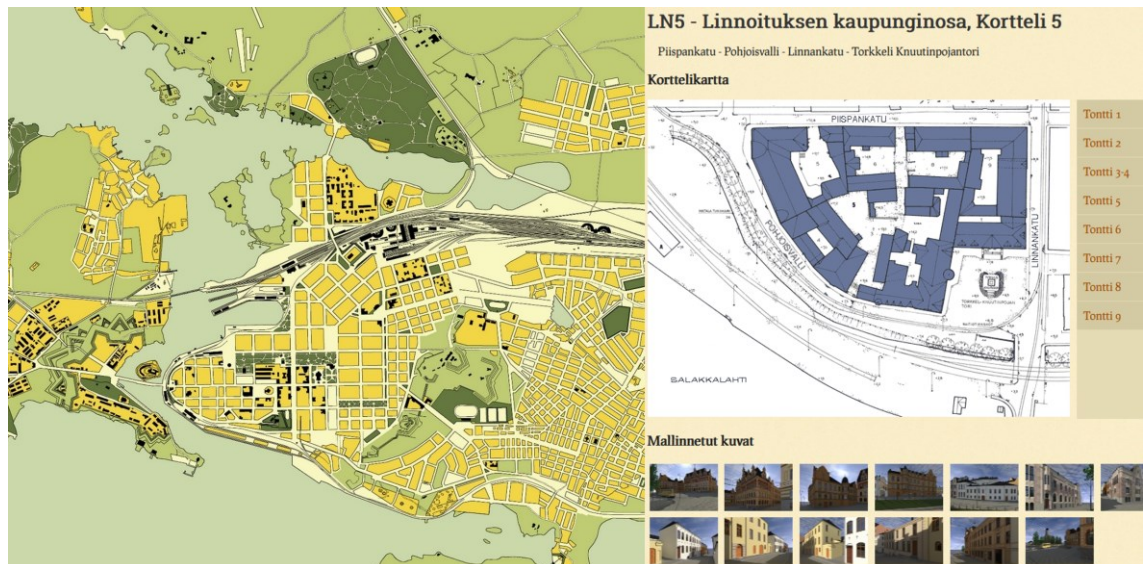
KUVA 7. Virtual Helsinki -kuva Senaatintorilta. Animaatiot on elävöitetty liikkuvilla ihmismalleilla. (Zoan 2022)

5 CASE VIRTUAALIVIIPURI

VirtuaaliViipuri-projekti on vuonna 2003 aloitettu tutkimus- ja kehityshanke. Projekti on tehty yhteistyössä Tampereen ammattikorkeakoulun kanssa ja sen kehitystyö on toteutettu opiskelijoiden toimesta. (VirtuaaliViipuri 2022)

VirtuaaliViipuri-projektin päätavoitteena on toteuttaa verkkosivulta saavutettava esitystapa historiallisesta Viipurista (KUVA 8). Lopullinen tavoite on virtuaalisesti liikuttava Viipuri. Projektin pääasiallinen toteutustapa on Viipurin rakennusten 3D-mallintaminen ja arkistointi. (VirtuaaliViipuri 2022)

Lähtöaineistona on käytetty intendentti Juha Lankisen tekemiä piirustuksia. Piirustukset ovat peräisin Lankisen suunnittelemasta Viipurin pienoismallista. Piirustukset käsittävät korkeuskäyrät, asemapiirustukset, rakennusten julkisivut ja värikaaviot. Lähtöaineistona on myös laaja arkisto valokuvia ja videoita Viipurista. (VirtuaaliViipuri 2022)



KUVA 8. VirtuaaliViipuri-verkkosivusto sisältää interaktiivisen kartan, joka toimii valikkona kortteleiden tarkempaa selaamista varten. (VirtuaaliViipuri 2022)

5.1 Lähtökohdat

Työn tavoitteena on toteuttaa yhtenäinen kaupunkimalli VirtuaaliViipuri-projektin mallinnuksia hyödyntämällä. Tavoitteena ei ole esimerkiksi uudelleen mallintaa projektin aikana tehtyjä rakennuksia vaan pyrkiä löytämään toimiva menetelmä mallinnusten hyödyntämiseen niiden nykytilassa.

VirtuaaliViipuri-projektin mallinnukset sijaitsevat projektin omassa tiedostoarkistossa, jonka koko on hieman ylitse yhden teratavun. Arkiston laajuus on siis kohtalaisen suuri, mikä vaikuttaa erityisesti hakusanalla tehtävän tiedon hakeamiseen. Tiedonkäsittely on työn toteuttamisen kannalta välttämätöntä, sillä joitain Viipurin rakennuksia on mallinnettu ja visualisoitu projektin yhteydessä useampaan kertaan. Mallinnuksia on myös useilta eri vuosilta, eikä yksittäisten mallinnusten viimeisimmille versioille ole omaa kansiotaan.

Projektin pääasiallisena mallinnustyökaluna on käytetty Archicad-ohjelmaa. Archicad on yleisesti arkkitehtien käyttämä rakennusten tietomallinnusohjelma. VirtuaaliViipuri-projektissa ei kuitenkaan hyödynnetä tietomalliominaisuuksia, sillä hanke ja arkistointimenetelmä perustuvat rakennusten visualisointiin historiallisella tarkkuudella.

BIM-menetelmät eroavat tavanomaisista peliteollisuuden menetelmistä, mikä vaikuttaa esimerkiksi suorituskykyyn reaaliaikaisissa sovelluksissa. Reaaliaikaisen suorituskyvyn vuoksi mallinnusten optimointi on välttämätöntä. Rakennusten tietomalleissa varsinainen mallinnus tapahtuu lähes aina rakennevahvuuksilla. Mallinnettu rakennus koostuu muun muassa erillisistä katto- ja seinäosista. Tietomallinnuksen seurauksena rakennuksiin muodostuu luonnostaan sisätila. Projektin mallinnukset voidaan siis luokitella CityGML-standardiin verraten tarkkuustasojen LOD3 ja LOD4 väliin. VirtuaaliViipuri-projektin tarkoituksena ei ole kuitenkaan esittää rakennusten sisätiloja, vaan pelkästään rakennusten ulko-kuori. Ylimääräinen geometria aiheuttaa teknisessä tarkastelussa noin neljä kertaa raskaamman mallinnuksen, kun verrataan vain tarpeellisen osan eli ulkokuoren esittämiseen (KUVA 9).



KUVA 9. Ote tietomallinnuksella toteutetusta seinärakenteesta verrattuna otteeseen pelkästä ulkokuoresta. (Varala 2022)

Osasta VirtuaaliViipuri-projektin mallinnetuista kortteleista on luotu Archicad-objekteja. Kortteliobjekteja on luotu esimerkiksi helpottamaan visualisointiprosessia. Yksittäisiä kortteliobjektitiedostoja on käteväämpää tuoda visualisointien taustalle kuin kortteleita Archicad-projektitiedostoista. Suuri osa kortteliobjekteista on myös aseteltu VirtuaaliViipuri-projektin Archicad-maastomalleihin, jolloin laajempia alueita voi tuoda visualisointeihin suoraan maastomallitiedostosta. Ongelmana on, että Archicad-ohjelma ei kykene avaamaan kymmeniä rakennuksia samanaikaisesti. Kortteleista on luotava Archicad-ohjelmaan tasoyhdistelmiä, joiden avulla voidaan avata sopiva määrä kortteleita kerrallaan.

5.2 Työvaiheet

Työ aloitetaan luomalla Unreal Engine -pelimoottoriin projektitiedosto. Unreal Engine -pelimoottori tarjoaa projektin luomisen yhteydessä vaihtoehtoja (template) valmiista pohjatiedostoista. Valittavana on pohjatiedostoja teeman mukaan muun muassa peleille, elokuville ja arkkitehtuurille. Pohjatiedostot sisältävät erilaisia asetuksia, joiden tehtävänä on nopeuttaa pelimoottorin käyttöönottoa käyttötarkoituksesta riippuen. Tässä työssä pohjatiedostoksi valikoituu arkkiteh-

tuuriteemainen yksinkertaistettu pohjatiedosto (Blank), joka sisältää esimerkiksi valmiiksi käyttöön otetun Datasmith-tiedostonkäsittelyjärjestelmän.

5.2.1 Aineiston yhdistäminen

Archicad-ohjelmaan asennetaan Datasmith-lisäosa, joka mahdollistaa mallinnusten siirtämisen ohjelmien välillä. Datasmith tuo tietomallinnusohjelmasta mukanaan käyttäjän luomat tasoasetukset ja esimerkiksi origon sijainnin suhteessa mallinnuksiin. Tämä mahdollistaa VirtuaaliViipuri-projektin maastomallitiedostoihin valmiiksi aseteltujen kortteliobjektien hyödyntämisen yhtenäisen kaupunkimallin luomisessa. Datasmith-tiedostoksi tallennetun kortteliobjektin sijainti suhteessa maastomalliin pysyy samana, vaikka ne tuotaisiin Unreal Engine -pelimoottoriin erillisinä tiedostoina.

Yksittäisiä kortteleita, joita ei ole sijoitettu VirtuaaliViipuri-projektin maastomallitiedostoon, ei voida tuoda suoraan oikealle paikalleen maastomalliin. Yksittäisiä kortteleita varten luodaan paikallinen tasokoordinaatisto Unreal Engine -projektiin Viipurin pohjakartan avulla. Pelimoottoriin tuotavat korttelit voidaan asettaa oikealle paikalleen Y- ja X-akseleilla pohjakartan mukaan ja korkeustaso Z-akselilla maastomallin mukaan.

5.2.2 Mallinnusten optimointi

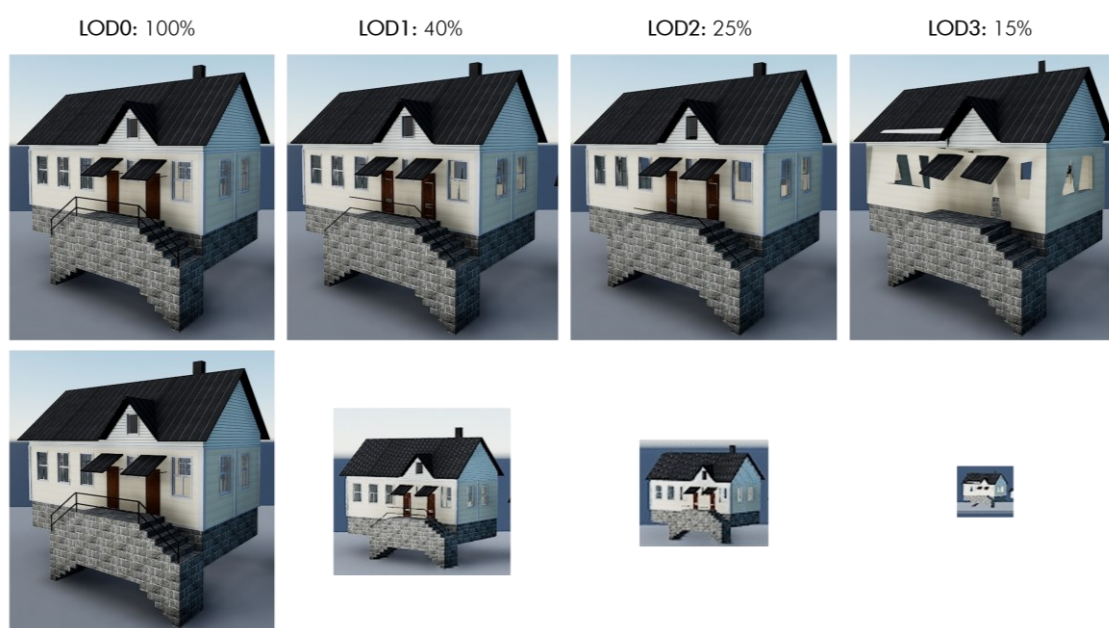
Työn yhteydessä tutkittiin mallien optimointia. Mallien optimoinnilla pyritään saavuttamaan reaaliaikaisesti tarkasteltava kaupunkimalli. Käytännössä rakennusmallien optimointi mahdollistaa sulavan työskentelyn Unreal Engine -pelimoottorissa.

Peliteollisuudessa tavanomainen optimointitapa yksittäisten mallinnusten tasolla on LOD-järjestelmän luominen mallinnuksesta. LOD-järjestelmällä tarkoitetaan optimointitekniikkaa, jossa mallinnuksesta luodaan eri tarkkuustason versioita ja nämä mallin tarkkuusversiot vaihtuvat käyttäjän asettaman virtuaalisen katseluetäisyyden mukaan. (Unreal Engine 2022j, Unity 2022, Godot 2022) Peliteollisuudessa LOD-tarkkuustasojen numerointi lasketaan iteraation mukaisesti eli

numerointi ilmenee päinvastaisena CityGML-standardiin verrattuna, jossa mallinnusten tarkkuus kasvaa numeroinnin kanssa.

Unreal Engine -pelimoottori sisältää työkalun LOD-tasojen automaattiseen luomiseen mallinnuksesta. Pelimoottorissa on mahdollista määrittää tarkkuustasojen määrä, yksittäisten tarkkuustasojen prosenttiosuus alkuperäisestä mallinnuksesta ja käyttöönoton etäisyys jokaiselle tarkkuustasolle.

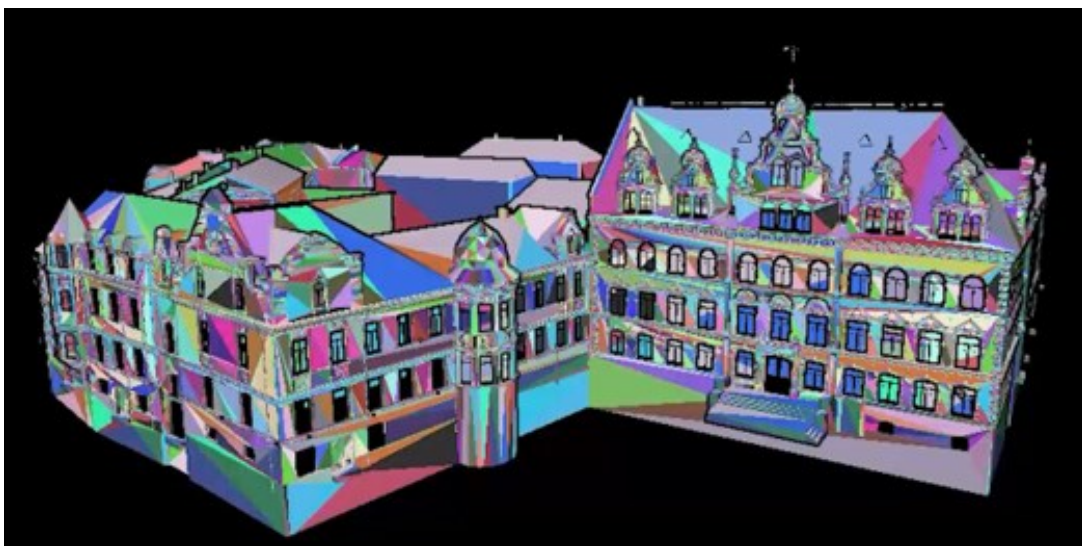
VirtuaaliViipuri-projektista tuodusta rakennuksesta luotiin pelimoottoriin automatisoidulla työkalulla kolme uutta tarkkuustasoa. Tarkastelussa huomataan, että LOD2-tarkkuustason mallilla ei ole merkittävää visuaalista eroa alkuperäiseen LOD0-tarkkuustason malliin (KUVA 10). Tämä vahvistaa lähtökohtaisen oletuksen rakennusmallin ylimääräisestä geometriasta pitävän paikkaansa.



KUVA 10. Tavanomainen LOD-järjestelmä havainnollistettuna BIM-menetelmällä toteutetussa rakennusmallissa. (Varala 2022)

Paras suorituskyky LOD-järjestelmästä saadaan tarkkuustasojen manuaalisella tuottamisella. Tarkkuustasojen manuaalinen luominen vaatii rakennusten mallintamista tietomalliohjelmistojen ulkopuolella ja tietomallinnuksesta eroavilla menetelmillä.

Toinen mallinnusten optimointitapa on Unreal Engine -pelimoottorin Nanite-järjestelmä. Nanite on täysin automatisoitu optimointitekniikka. Nanite-järjestelmä päättelee geometrian tarkkuustason ja pyrkii piilottamaan näkymän taakse jäävää geometriaa reaaliaikaisesti (KUVA 11 ja 12). (Unreal Engine 2022g) Nanite-järjestelmä vähentää merkittävästi VirtuaaliViipuri-projektin mallinnuksista aiheutuvaa manuaalista optimointityötä.



KUVA 11. Virtualisoidun geometrian visualisointi Unreal Engine -pelimoottorissa. Polygonit on esitetty eri väreillä. (Varala 2022)



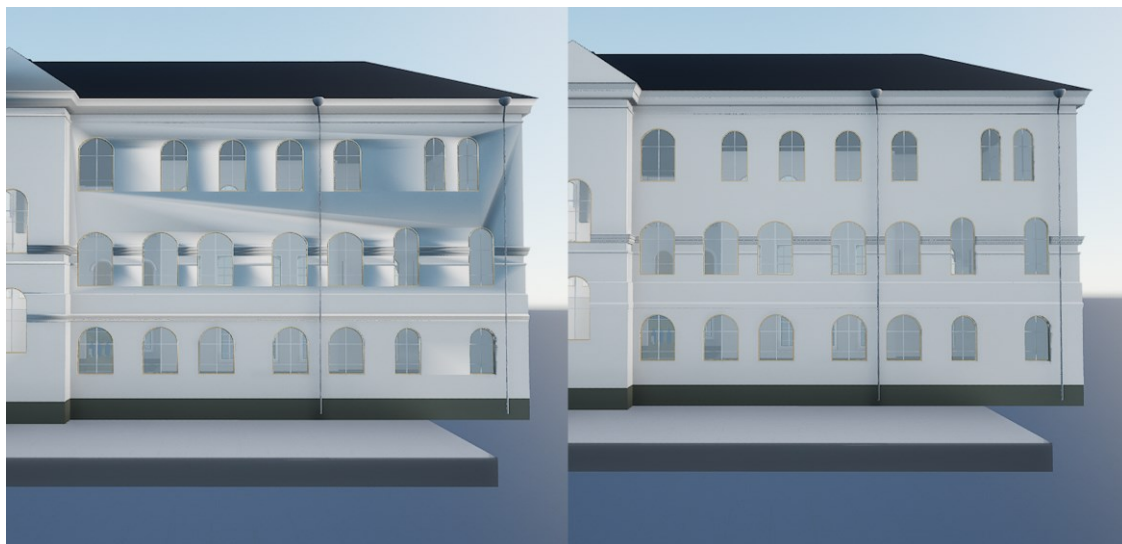
KUVA 12. Polygonien päällekkäispiirron (overdraw) visualisointi Unreal Engine -pelimoottorissa. (Varala 2022)

VirtuaaliViipuri-projektissa toteutettu yksittäinen mallinnus saattaa sisältää kymmeniä erillisiä tekstuuritiedostoja. Tekstuuritiedostot esittävät mallinnuksien pinnoilla tietyt materiaaleja ja värejä. Tekstuurien määrästä huolimatta niiden ei katsota vaikuttavan merkittävästi kaupunkimallin reaaliaikaiseen suorituskyyntä pelimoottorissa. Tämä päätelmä johtuu tekstuurikuvien merkittävän pienestä resoluutiosta ja tiedostokoosta.

5.3 Ongelmat ja ratkaisut

Työssä esiintyvät tekniset ongelmat vaikuttavat pääasiassa VirtuaaliViipuri-projektin mallinnuksia hyödyntämällä tuotetun kaupunkimallin visuaaliseen tasokkuuteen. Ongelmien ratkaiseminen on tärkeää visuaalisesti korkeatasoisen kaupunkimallin tuottamiseksi.

Työssä hyödynnettyjen kortteliobjektien pinnoilla esiintyy satunnaisesti varjotusvirheitä (KUVA 13). Varjotusvirheet on mahdollista korjata suoraan Unreal Engine -pelimoottorissa. Korjaus tapahtuu automaattisesti Unreal Engine -muokkausvalikosta napsauttamalla pintojen uudelleenlaskenta-kohtaa (recompute normals) ja hyväksymällä muutokset (apply changes).



KUVA 13. Ennen ja jälkeen kortteliobjektin pinnan varjotusvirheiden korjaamisen. (Varala 2022)

Nanite-optimointijärjestelmän käyttöönoton yhteydessä havaittiin muutamia visuaalisia ongelmia. Näitä ongelmia olivat muun muassa läpinäkyvät pinnat ja

tummentuneet materiaalit mallinnuksien pinnoilla (KUVA 14 ja 15). Ongelmien korjaaminen Unreal Engine -pelimoottorissa on mahdollista, mutta aiheuttaa manuaalisen työmäärän kasvua mallin käsittelyssä ja korjaamisessa. Esimerkiksi tummentuneet materiaalit korjataan muuttamalla Unreal Engine -materiaalimuokkausvalikosta jokaisen materiaalien kaksipuolisuus-asetusta (two-sided). VirtuaaliViipuri-projektin mallinnuksissa materiaalilla tarkoitetaan tekstuuritiedostoja, joita yksittäisissä mallinnuksissa on useita. Asetuksien säätäminen on mahdollista automatisoida tuonnin yhteydessä, mutta tämä vaatii käsittelytavan ohjelmoinnin pelimoottorin Dataprep-työkalulla.



KUVA 14. Ennen ja jälkeen Nanite-järjestelmään liittyvän kaksipuolisuusongelman (two-sided) korjaamisen. (Varala 2022)



KUVA 15. Ennen ja jälkeen Nanite-järjestelmän käyttöönoton. Materiaalit esiintyvät tummina ja virheellisinä. (Varala 2022)

6 POHDINTA

Työn tuloksena syntyi yhtenäinen kaupunkimalli historiallisesta Viipurista. Kaupunkimalli koottiin Unreal Engine -projektitiedostoksi, joka toimii alustana VirtuaaliViipuri-projektin ja kaupunkimallin mahdolliselle jatkokehitykselle. Kaupunkimalli ei sisällä kaikkia VirtuaaliViipuri-projektin mallinnuksia, vaan siihen on sisällytetty ainoastaan Archicad-maastomallitiedostoihin tuodut kortteliobjektit. Tämä johtuu kaupunkimallin täydentämiseen liittyvästä työmäärästä, jota ei ollut mahdollista suorittaa opinnäytetyön aikana.



KUVA 16. Näytönkaappauskuva Unreal Engine -pelimoottorin editorissa reaaliaikaisesti tarkasteltavasta kaupunkimallista. (Varala 2022)

Kaupunkimallia voidaan jatkokehittää täydentämällä Unreal Engine -projektiin yksittäisiä kortteleita VirtuaaliViipuri-projektin arkistosta. Täydentämisen lisäksi myös kaupunkimallin visuaalisia ominaisuuksia voidaan parantaa muun muassa tekstuurien päivittämisellä ja yksittäisten virheiden korjaamisella. Kaupunkimallia voidaan elävöittää Unreal Engine -pelimoottorin yhteydessä käytettävissä olevilla materiaalikirjastoilla kuten Quixel Bridge -kirjastolla, joka tarjoaa fotorealistisia puita ja pintatekstuureja.

Unreal Engine -pelimoottorialustainen Viipurin kaupunkimalli mahdollistaa merkittävää joustavuutta sovelluskehitykselle. Kaupunkimallista voidaan esimerkiksi toteuttaa 360-kuvia, kuten Digimuseon Turku Åbo 1827 -virtuaalinäyttelyssä (KUVA 17) (Digimuseo 2022). Tällaiseen virtuaaliseen kokemukseen olisi mahdollista liittää aineistoa myös VirtuaaliViipurin verkkosivuilta, jolloin esitystapa olisi nykyistä verkkosivua osallistavampi käyttäjäkokemus. Googlen katunäkymäpalvelun kaltainen sovellus Viipurin kaupunkimallista olisi myös helposti saatettava laajalle yleisölle.



KUVA 17. Digimuseon virtuaalinäyttelyn käyttöliittymä selaimessa. (Digimuseo 2022)

Unreal Engine -pelimoottori saattaa tulevaisuudessa kasvattaa suosiotaan kaupunkimallien sovelluksissa. Pelimoottori tarjoaa nykytilassaan laajan valikoiman ominaisuuksia, jotka edistävät sen yhteistoimintaa suunnittelualan työmenetelmien kanssa. Opinnäytetyössä käytetyn Unreal Engine -pelimoottorin uusimman versio (5.0.0) on työn toteuttamisen ajankohtana vielä melko uusi julkaisu, joten pelimoottorissa esiintyvät ohjelmistovirheet kuten Nanite-järjestelmän ongelmat pintojen kanssa tullaan tulevaisuudessa korjaamaan.

LÄHTEET

- Airaksinen, E. 2017. Diplomityö. Kaupunkien kolmiulotteiset mallinnusmenetelmät. Luettu 2.5.2022. <http://urn.fi/URN:NBN:fi:aalto-201706135482>
- Alatalo, T., Koskela, T., Pouke, M., Alavesa, P. & Ojala, T. 2016. VirtualOulu: Collaborative, Immersive and Extensible 3D City Model on the Web. Luettu 12.5.2022. <http://dx.doi.org/10.1145/2945292.2945305>
- Barczak, A. & Woźniak, H. 2019. Comparative Study on Game Engines. Luettu 28.4.2022. <https://doi.org/10.34739/si.2019.23.01>
- Biljecki, F., Stoter, J., Leadoux, H., Zlatanova, S. & Coltekin, A. 2015. Luettu 25.4.2022. Applications of 3D City Models: State of the Art Review. <https://www.mdpi.com/2220-9964/4/4/2842/>
- Billen, R., Cutting-Decelle, A.F., Marina, O., de Almeida, J.P., Matteo, C., Falquet, G., Leduc, T., Métral, C., Moreau, G. & Perret, J. 2014. 3D City Models and urban information: Current issues and perspectives. Luettu 25.4.2022. https://www.researchgate.net/publication/260172519_3D_City_Models_and_urban_information_Current_issues_and_perspectives
- Cohen-Or, D., Chrysanthou, Y. & Silva, C-T. A Survey of Visibility for-Walkthrough Applications. Viitattu 17.5.2022. https://www.researchgate.net/publication/2440562_A_Survey_of_Visibility_for_Walkthrough_Applications
- Digimuseo. 2022. Verkkosivu. Viitattu 18.5.2022. <https://digimuseo.fi/>
- Freiknecht, J., Geiger, C., Drochert, D., Effelsberg, W. & Dörner, R. 2016. Game Engines. Luettu 28.4.2022. https://doi.org/10.1007/978-3-319-40612-1_6
- Godot. 2022. Verkkosivu. Viitattu 23.5.2022. <https://godotengine.org/article/dev-snapshot-godot-4-0-alpha-1>
- Julin, A., Jaalama, K., Virtanen, J-P. & Airaksinen, E. 2020. Emerging Web and Game Engine Tech for 3D Cities. Luettu 12.5.2022. <https://www.gim-international.com/content/article/emerging-web-and-game-engine-tech-for-3d-cities>
- Julin, A., Jaalama, K., Virtanen, J-P., Pouke, M., Ylipulli, J., Vaaja, M., Hyypä, J. & Hyypä, H. Characterizing 3D City Modeling Projects: Towards a Harmonized Interoperable System. 2018. Luettu 26.4.2022. <https://doi.org/10.3390/ijgi7020055>
- Open Geospatial Consortium (OGC). 2012. OGC City Geography Markup Language (CityGML) Encoding Standard. Viitattu 13.5.2022. <http://www.opengis.net/spec/citygml/2.0>

Tampereen kaupunki. 2022. Koordinaati- ja korkeusjärjestelmät. Viitattu 23.5.2022. <https://www.tampere.fi/tampereen-kaupunki/tietoa-tampereesta/kartat/koordinaatti-ja-korkeusjarjestelmat.html>

Tampereen kaupunkimittaus. 2018. Tampereen kaupungin Unity 3D kaupunkimalli. Viitattu 23.5.2022. <https://data.tampere.fi/data/fi/dataset/tampereen-kaupungin-unity-3d-kaupunkimalli>

Unity. 2022. Verkkosivu. Viitattu 23.5.2022. <https://docs.unity3d.com/Manual/LevelOfDetail.html>

Unreal Engine. 2022a. Verkkosivu. Viitattu 4.5.2022. <https://www.unrealengine.com/en-US/>

Unreal Engine. 2022b. Verkkosivu. Viitattu 4.5.2022. <https://www.unrealengine.com/en-US/eula/unreal>

Unreal Engine. 2022c. Verkkosivu. Viitattu 4.5.2022. <https://docs.unrealengine.com/5.0/en-US/datasmith-plugins-for-unreal-engine/>

Unreal Engine. 2022d. Verkkosivu. Viitattu 4.5.2022. <https://docs.unrealengine.com/5.0/en-US/dataprep-import-customization-in-unreal-engine/>

Unreal Engine. 2022e. Verkkosivu. Viitattu 4.5.2022. <https://docs.unrealengine.com/5.0/en-US/lidar-point-cloud-plugin-for-unreal-engine/>

Unreal Engine. 2022f. Verkkosivu. Viitattu 4.5.2022. <https://docs.unrealengine.com/5.0/en-US/world-partition-in-unreal-engine/>

Unreal Engine. 2022g. Verkkosivu. Viitattu 4.5.2022. <https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/>

Unreal Engine. 2022h. Verkkosivu. Viitattu 4.5.2022. <https://docs.unrealengine.com/5.0/en-US/lumen-global-illumination-and-reflections-in-unreal-engine/>

Unreal Engine. 2022i. Verkkosivu. Viitattu 4.5.2022. <https://www.unrealengine.com/en-US/bridge>

Unreal Engine. 2022j. Verkkosivu. Viitattu 4.5.2022. <https://docs.unrealengine.com/5.0/en-US/creating-and-using-lods-in-unreal-engine/>

Virtanen, J-P., Julin, A., Hyypä, H., Ahlavo, M. & Hyypä, J. 2017. Tulevaisuuden avoimet kaupunkimallit. Maankäyttö 4/2017. Luettu 12.5.2022. http://www.maankaytto.fi/arkisto/mk417/mk417_2015_virtanen_julin_et_al.pdf

VirtuaaliViipuri. 2022. Verkkosivu. Viitattu 4.5.2022. <https://virtuaaliviipuri.fi/fi>

Zoan. 2022. Verkkosivu. Viitattu 23.5.2022 <https://zoan.fi/cases/virtual-city-helsinki/>