

Olli-Pekka Koverola

KARTTAPOHJAINEN  
KAUPUNKIMALLINNUS  
CITYENGINELLÄ

Opinnäytetyö  
Tietojenkäsittelyn koulutusohjelma

Kesäkuu 2014




**MIKKELIN AMMATTIKORKEAKOULU**

Mikkeli University of Applied Sciences

## KUVAILULEHTI

 <b>MIKKELIN AMMATTIKORKEAKOULU</b> <small>Mikkeli University of Applied Sciences</small>	<b>Opinnäytetyön päivämäärä</b>  19.6.2014	
<b>Tekijä(t)</b>  Olli-Pekka Koverola	<b>Koulutusohjelma ja suuntautuminen</b>  Tietojenkäsittelyn koulutusohjelma	
<b>Nimeke</b>  Karttapohjainen kaupunkimallinnus Cityenginellä		
<b>Tiivistelmä</b>  Kolmiulotteisella kaupunkimallilla tarkoitetaan kaupunkimallia, joka on yleensä mallinnettu kaupungin nykytilasta. Malleja voidaan käyttää muun muassa erilaisten projektien havainnollistamiseen tai uusien suunnitteluun. Tämän opinnäytetyön tarkoituksena on selvittää, kuinka CityEngine sopii rakennusten mallintamiseen.  Teoriaosuudessa käytiin ensimmäiseksi läpi mallintamisen perusidea, minkä jälkeen siirryttiin itse kaupunkimalleihin ja aineistoihin. Tässä osiossa perehdyttiin muun muassa siihen, mitä kaupunkimallit ovat ja minkälaisiin tarkoituksiin niitä voidaan käyttää. Tämän lisäksi teoriaosuuteen kuului vielä kaupunkimallinnukseen tarkoitettujen aineistojen tutkiminen sekä perehtyminen työssä käytettyyn ohjelmistoon.  Työn aikana perehdyttiin muun muassa siihen, kuinka OpenStreetMap-karttapohja sopii työssä käytettyyn ohjelmaan sekä minkälaisia menetelmiä ohjelma tarjoaa rakennusten mallintamiseen. Työvaiheessa kerättiin talteen omia ajatuksia ohjelman soveltuvuudesta sekä erilaisista mallinnusmenetelmistä. Loppuen lopuksi selvisi, että erilaisia menetelmiä oli muutamia. Sääntöihin perustuva mallinnusmenetelmä sopii omasta mielestä niin rakennuksien kuin isompien alueiden mallintamiseen. Erittäin tarkkojen mallien mallintamiseen käyttäisin esimerkiksi SketchUpia ja ohjelman polygonal shape creation -työkalua hyvin yksinkertaisten muotojen tekemiseen.		
<b>Asiasanat (avainsanat)</b>  3D, kaupunkimalli, kaupunkimallinnus, CityEngine		
<b>Sivumäärä</b>  40	<b>Kieli</b>  Suomi	<b>URN</b>
<b>Huomautus (huomautukset liitteistä)</b>		
<b>Ohjaavan opettajan nimi</b>  Esa Hannus	<b>Opinnäytetyön toimeksiantaja</b>  Mikkelin ammattikorkeakoulu	

## DESCRIPTION

 <p><b>MIKKELIN AMMATTIKORKEAKOULU</b> Mikkeli University of Applied Sciences</p>		<b>Date of the bachelor's thesis</b>  19 <sup>th</sup> of June 2014
<b>Author(s)</b>  Olli-Pekka Koverola	<b>Degree programme and option</b>  Business Information Technology	
<b>Name of the bachelor's thesis</b>  Map based urban modelling with CityEngine		
<b>Abstract</b>  <p>Usually three dimensional urban models are based on real cities. These models can be used for demonstrating projects or planning new ones. The purpose of this bachelor's thesis was to study how CityEngine worked for building modelling purposes and what kind of methods this program offered for modelling. The theoretical part of this thesis presented the basic modelling methods and also urban models. The theoretical part also introduced the program that was used in this thesis and some other modelling software.</p> <p>The practical part of this thesis was to study what kind of methods CityEngine offered for modelling buildings and how to use street network data which was imported from an OSM file. The main focus of this thesis was to demonstrate different modelling methods and how to import the OSM file into the CityEngine scene. The buildings that were modelled using these methods were the school buildings of Mikkeli university of applied sciences. Finally I presented how to add CityEngine models to web browser using software own tools.</p> <p>It appeared that CityEngine offered few modelling methods. In my opinion some of the methods were better than other methods. For example, I would not use CityEngine drawing methods for detailed buildings. Instead, I would use SketchUp for that purpose, because it is a lot easier to model high quality buildings in SketchUp than in CityEngine. The OSM file that was exported from the OpenStreetMap site was easy to use and it created a good map layer for the CityEngine scene.</p>		
<b>Subject headings, (keywords)</b>  3D, CityEngine, urban models, city modeling		
<b>Pages</b>  40	<b>Language</b>  Finnish	<b>URN</b>
<b>Remarks, notes on appendices</b>		
<b>Tutor</b>  Esa Hannus	<b>Bachelor's thesis assigned by</b>  Mikkeli University of Applied Sciences	

# SISÄLTÖ

1	JOHDANTO .....	1
2	3D KAUPUNKIMALLI .....	2
2.1	3D-mallintaminen .....	2
2.1.1	3D-mallinnus .....	3
2.1.2	Level of detail .....	4
2.1.3	3D-mallintamisen käyttötarkoitus .....	6
2.2	Kaupunkimalli .....	7
2.2.1	Kaupunkimallin käyttötarkoitus .....	8
2.2.2	Kaupunkimallien tulevaisuus .....	9
2.3	Kaupunkimallin aineistot .....	10
2.3.1	Paikkatieto .....	10
2.3.2	Paikkatietoaineisto .....	11
2.3.3	Kartat, kuva-aineistot ja Internet-Karttapalvelut .....	11
2.3.4	Openstreetmap .....	16
2.4	Ohjelmistot kaupunkimallinnukseen .....	17
2.4.1	Esrin CityEngine .....	18
2.4.2	ArcGIS ja 3D Analyst .....	20
2.4.3	Muita kaupunkimallinnusohjelmistoja .....	21
3	KAMPUKSEN MALLINTAMINEN CITYENGINELLÄ .....	23
3.1	OpenStreetMap & CityEngine .....	24
3.1.1	Sääntöihin perustuva mallinnus .....	26
3.1.2	Kuvan perusteella mallintaminen .....	31
3.1.3	Mallintaminen piirtämällä .....	33
3.2	Web viewer .....	37
4	YHTEENVETO .....	38
	LÄHTEET .....	41

## 1 JOHDANTO

Kolmiulotteiset kaupunkimallit ovat erityisen hyvä tapa esittää uusia aluesuunnitteluita. Tekniikan kehittyessä kaupunkimallien tekeminen ei enää vaadi kuukausia, vaan yksinkertaisemman kaupungin tai kaupunginosan pystyy rakentamaan jopa muutamassa päivässä. Mallien avulla voidaan muun muassa havainnollistaa rakennusten vaikutukset ja haitat ympäristölle. Tällaisten ongelmien havaitseminen jo suunnitteluvaiheessa auttaa säästämään huomattavan määrän rahaa ja aikaa.

Tämän opinnäytetyön tarkoituksena on selvittää kuinka kaupunkimallinnusohjelma CityEngine soveltuu Mikkelin ammattikorkeakoulun kampuksen mallintamiseen sekä minkälaisia erilaisia menetelmiä ohjelma tarjoaa rakennusten mallintamiseen. Samalla tutkin kuinka OpenStreetMap-aineistoa voidaan käyttää ohjelman kanssa. Lopuksi on vielä tarkoitus selvittää kuinka kolmiulotteisen mallin vieminen onnistuu selaimelle käyttäen ohjelman Web Viewer -toimintoa. Tämä työ on osa koulun paikkatietojärjestelmän ja 3D-mallinnusopetuksen kehitystä.

Yleisesti ottaen kaupunkimalli koostuu maaston ja rakennusten mallintamisesta. Ennen kuin rakennuksia tai maastoja voidaan edes mallintaa, on tiedettävä kolmeulotteisesta mallintamisesta perusasioita. Tämän jälkeen voidaan miettiä erilaisia aineistoja kaupunkimallin luomiseen. Vasta näiden jälkeen, voidaan miettiä ohjelmistoa kaupunkimallin luomiseen. Perehdyn näihin asioihin yleisellä tasolla sekä esittelen joitakin ohjelmistoja hieman muita laajemmin. Kaikkia asioita ja ohjelmistoja en tässä työssä kuitenkaan käy läpi vaan ainoastaan ne, jotka ovat jollakin tapaa osana tai lähellä tätä työtä. Teoriaosuuden jälkeen kerron vielä työstä ja lopuksi yhteenvedossa esittelen työssä esiintyneitä ongelmia ja omia mielipiteitä sekä mietin mahdollisia jatkokehitysideoita.

## 2 3D-KAUPUNKIMALLI

Kolmiulotteiset kaupunkimallit ovat varmasti tuttuja kaikille, jotka ovat joskus pelanneet tietokonepelejä tai edes niitä seuranneille. Peleihin usein mallinnetaan kuvitteellisia kaupunkeja, mutta pääsääntöisesti malleja tehdään oikeista kaupungeista. Silloin niiden tarjoamat edut ovat parhaimmillaan. Kun malli tehdään oikein ja hyvin, saadaan siitä kaupungilla tehokas ja kustannuksia säästävä työkalu. Oikeista kaupungeista tehtyjä kaupunkimalleja onkin tehty paljon ympäri maailmaa ja mukaan lukien Suomessa, mutta Suomessa vain vähän. Vaikka kaupunkimalleja onkin muualla paljon, eivät ne välttämättä tarjoa kaikkea mitä ne voisivat tarjota.

"Muualla Euroopassa, esimerkiksi Ranskassa, kaupunkimalleja on paljon. Monet niistä ovat kuitenkin vain näyttäviä visualisointeja, joita ei voi hyödyntää suunnittelun lähtötietona ainakaan kovin monipuolisesti", toteaa Luoma Vianova Systems Oy:stä Position-lehdelle (Isotalo 2013, 17).

Ennen kuin kolmiulotteisista kaupunkimalleista voitaisiin puhua lisää, on tiedettävä, mitä kolmiulotteisella mallintamisella tarkoitetaan. Kolmiulotteinen mallintaminen on kuitenkin hyvin iso osa-alue kaupunkimallia. Siksi aloitan seuraavaksi käymällä läpi hieman perusasioita 3D-mallintamisesta.

### 2.1 3D-mallintaminen

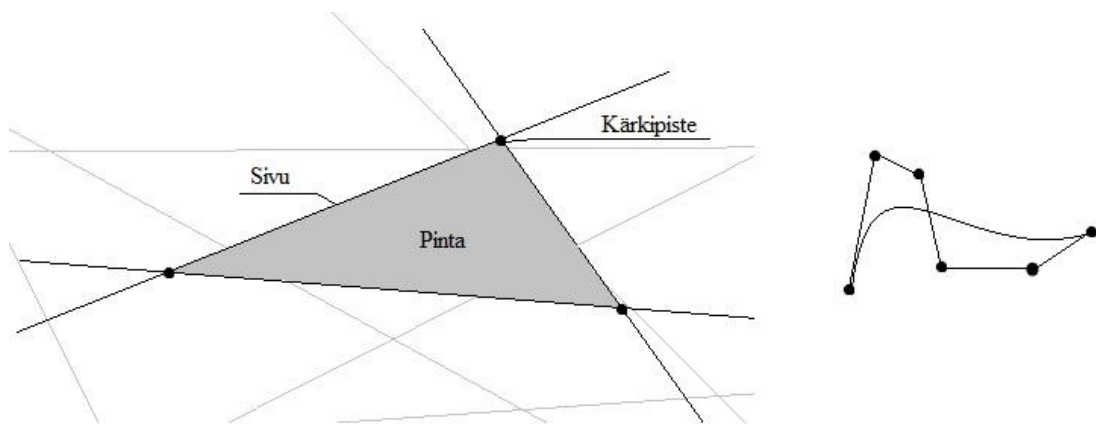
Lyhyesti 3D-mallintamisella tarkoitetaan erilaisten kappaleiden suunnittelua kolmiulotteisesti. Kyseessä siis voi olla tavallinen kahvimuki tai kokonainen kaupunki, joka on mallinnettu kolmiulotteiseen malliavaruuteen. Suunnittelijoille 3D-mallinnus tarjoaa erinomaisen menetelmän tuotteen suunnitteluun ja sen havainnollistamiseen asiakkaalle.

Aloitan tämän työn käymällä ensimmäiseksi läpi perusasioita 3D-mallintamisesta. Kerron muun muassa seuraavissa kappaleissa mitä tarkoitetaan 3D-mallinnuksella ja LOD-tasoilla (level of detail) sekä LOD-tasojen merkityksen kaupunkimallinnuksessa. Näiden jälkeen käyn vielä läpi suunnittelijoiden näkökulmasta, miksi 3D-mallinnus on erinomainen menetelmä erilaisissa kehitys- ja suunnittelutöissä sekä sen kannattavuuden erilaisissa projekteissa.

### 2.1.1 3D-mallinnus

3D-mallinnuksella tarkoitetaan kappaleiden suunnittelua kolmiulotteisesti. Malli tehdään kolmiulotteisessa avaruudessa, joka koostuu x-, y-, z-koordinaattiakseleista. Mallinnettava kappale on mahdollista mallintaa aivan oikean kappaleen näköiseksi sekä sille on myös mahdollista lisätä kaikki fyysiset ja mekaaniset ominaisuudet, jotka oikeallakin kappaleella olisi. Mallinnettaessa samoja muotoja kappale mallinnetaan vain kerran jonka jälkeen kappaletta kopioidaan tarvittava määrä. Mallia mallinnettaessa on myös hyvä ottaa huomioon mallin mahdolliset materiaalit sekä animoidaanko malli vai ei. (Lehtovirta & Nuutinen 2000, 22; Tuhola & Viitanen 2008, 17.)

3D-mallilla tarkoitetaan kolmiulotteista kappaletta, jota on mahdollista tutkia kolmiulotteisessa avaruudessa. Malli rakennetaan usein mesh-verkosta, joka taas koostuu pinnoista (face), sivuista (edge) ja kärkipisteistä (vertex). Verkosto siis koostuu pinnoista, jotka jakavat vähintään yhden kärkipisteen naapuripinnan kanssa kuten kuvassa 1 nähdään vasemmalla. (Lehtovirta & Nuutinen 2000, 21; Tuhola & Viitanen 2008, 20.)

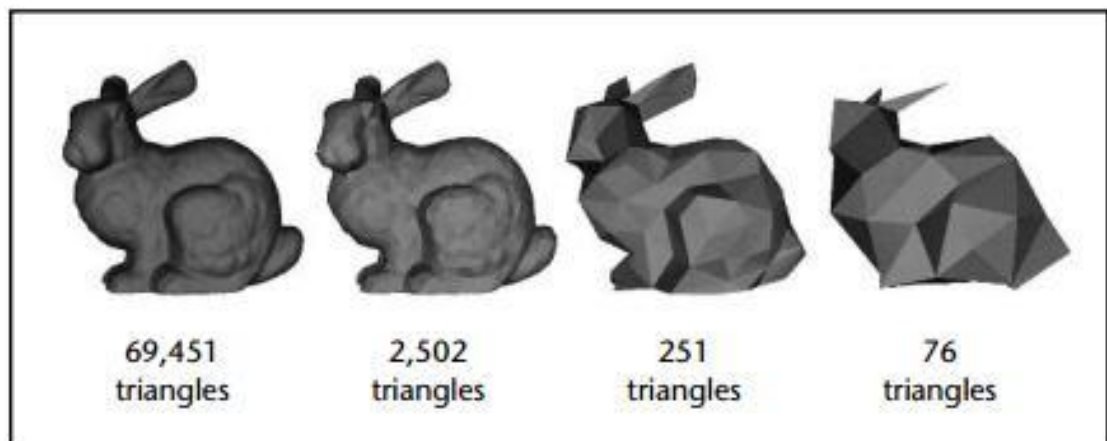


**KUVA 1. Mesh-verkon osat vasemmalla ja oikealla NURBS-käyrä**

Mesh-verkon lisäksi on olemassa myös muitakin mallinnuskeinoja. Yksi näistä keinoista on NURBS-mallinnus (Non-Uniform Rational B-splines). NURBS-mallinnuksessa kappale rakennetaan spline-käyrillä, joiden välillä syntyy kappaleen pinta. NURBS-käyrä on myös esitetty kuvassa 1. (Lehtovirta & Nuutinen 2000, 21.)

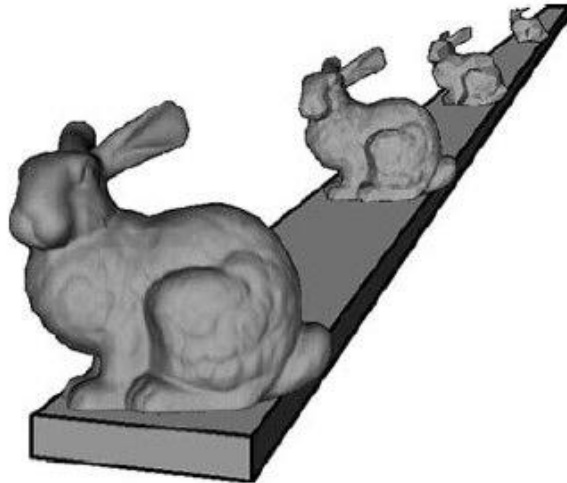
### 2.1.2 Level of detail

Tarkkuustasojärjestelmä, joka yleisemmin tunnetaan level of detail (LOD) -järjestelmänä, on laajasti käytetty järjestelmä 3D-mallinnuksessa ja erityisesti kaupunkimallien luomisessa. Tämän järjestelmän ansiosta 3D-malleja on mahdollista esittää huomattavasti kevyemmin. Tarkkuustasojärjestelmän ideana on asettaa kohdemallille tarvittava määrä muotoja ja tekstuureja tietylle katseluetäisyydelle. Kun kohdetta tarkastellaan kaukaa, on turhaa lisätä mallille mitään yksityiskohtia, sillä niitä ei kuitenkaan näkyisi. Usein suuremmassa näkymässä on paljon enemmän kohteita ja mitä vähemmän kohteilla on pintoja, sen kevyemmin kuvaa voidaan käsitellä. Lähemmäksi taas mentäessä kohteiden määrä vähenee, jolloin tarkastettavalle kohteelle voidaan lisätä enemmän pintoja ja näin ollen saadaan mallista paljon tarkempi ja yksityiskohtaisempi kuva. Kuvassa 2 nähdään neljä erilaista LOD tasoa, jossa vasemmalta oikealle mentäessä monikulmioiden (polygon) määrä vähenee. (Erving 2007, 61; Luebke ym. 2002, 5–6)



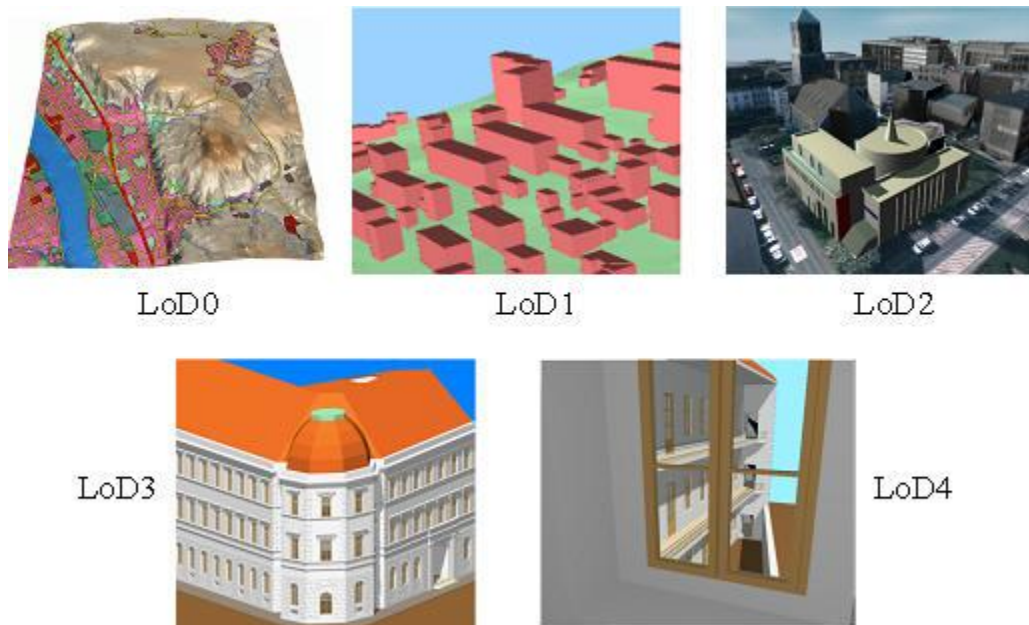
**KUVA 2. Neljä erilaista LOD-tasoa (Luebke ym. 2002)**

Pelimoottorit hyödyntävät jatkuvasti LOD-tasoja, sillä tasojen avulla saadaan ympäristöstä paljon kevyemmin toimiva ilman, että karsitaan yksityiskohtia. Idea toimii siten, kun pelimaailmassa edetään, näytetään ainoastaan kameraa lähellä olevat objektit tarkasti ja kaukana olevat epätarkasti. Kameran lähestyessä kauimmaista objektia objektin tarkkuus paranee ja taakse jääneen objektin tarkkuus huononee. Kuvassa 3 on esitetty kohteen tarkkuus katseluetäisyydestä. (Kuhno 2012, 6.)



**KUVA 3. Jänikset asetettu katseluetaäisyyksiin siten, että kaikista tarkin malli on ensimmäisenä ja viimeisenä epätarkin malli (Luebke ym. 2002)**

Kaupunkimalleissa käytetään yleensä viittä erilaista LOD-tasoa. Nämä tasot perustuvat kaupunkimallinnuksessa yleensä CityGML-standardiin. Standardi määrittelee tarkasti kuinka rakennukset tulee kuvata missäkin LOD-tasossa. Seuraavaksi esitetään CityGML-standardin mukaiset LOD-tasot sekä kuvassa 4 on havainnollistettu kyseiset tasot.



**KUVA 4. Viisi erilaista LOD-tasoa jotka perustuvat CityGML standardiin (Gröger ym. 2008)**

LOD0 on tasoista karkein ja jota käytetään kaksi- ja puoliulotteisen (2,5D) digitaalisen maastomallin kuvaamiseen. Nolla tasossa näytetään joko ilmakuvaa tai karttakuvaa alueesta. LOD1-tasossa näytetään talot laatikkomaisina ja ne ovat nostettu rakennuksen karkeaan korkeuteen. LOD2 on Euroopassa käytetyin tarkkuustaso. Kyseisellä tasolla voidaan muun muassa tehdä varjostus-, melu ja energia-analyysit. Tasossa talot ovat edelleen laatikkomaisia, mutta niihin on jo lisätty erilaisia kattogeometrioita, ja rakennus voi sisältää muitakin erimuotoisia pintoja. Rakennuksen tulee jo näyttää tekstuuriltaan itseltään sekä kuvaus kasvillisuudesta saattaa sisältyä malliin. LOD3-tasossa rakennus on hyvin yksityiskohtainen ja joka mm. sisältää kattorakenteita, parvekkeita, räystäitä jne. Lisäksi rakennuksessa pitää olla fotorealistiset tekstuurit. Samaan tasoon kuuluu myös ympäristön, kasvillisuuden ja liikenteen yksityiskohtainen kuvaus. Kaupunkimallinnuksen tasoista viimeisin ja tarkin taso LOD4 sisältää LOD3-tason lisäksi yksityiskohtia rakennuksen sisäpuolelta. Sisätilojen kuvaamiseen kuuluvat muun muassa ovet, ikkunat, sekä portaat. (Gröger ym. 2008, 9; Suomisto 2013.)

### 2.1.3 3D-mallintamisen käyttötarkoitus

Hyvin tehty 3D-malli on monipuolinen ja edullinen tapa esitellä tuote muille henkilöille tai asiakkaille. Kolmiulotteisen mallin mallintaminen on huomattavasti nopeampaa ja halvempaa kuin itse tuotteen rakentaminen. Vaikka kyseessä on vain malli eikä oikea kappale, niin sillä voidaan silti esitellä vaikeammatkin asiat helposti muille. 3D-malli on kuitenkin parempi kuin pelkän ajatuksen esittäminen.

Lehtovirta ja Nuutinen (2000, 11) toteavat kirjassaan, että ”Perinteisesti kolmiulotteinen suunnittelu on ollut vahvasti sidoksissa ’tekniseen’ suunnittelumaailmaan”. He jatkavat vielä, että ”Nykyään, kuten kaikissa projekteissa, joissa kehitys kulkee lujaa eteenpäin, on myös mallinnustekniikkaa sovellettu pidemmän aikaa monille tuotantoalueille”, ja näinhän tämä on ollut viimeiset 14 vuotta. (Lehtovirta & Nuutinen 2000, 11.) Kolmiulotteista mallintamista käytetään jatkuvasti esimerkiksi erilaisissa kaupunkihankkeissa, tuotesuunnitteluissa, pelimaailmoissa ja elokuvissa.

3D-mallinnus näyttölee jo tänä päivänä isoa roolia tuotteiden markkinoinnissa ja tuotekehittelyssä. Juurikin sen nopeuden, rajojen ja edullisuuden vuoksi se on erinomainen menetelmä tuottaa sekä suunnitella tuotteita. Esimerkiksi kuvitellaan uuden auton rakentamista. On huomattavasti halvempaa ja nopeampaa tehdä suunnitelmasta 3D-

malli kuin rakentaa itse auto ja todeta, että se ei ole hyvä. Eihän se tietenkään vastaa todellisuutta, mutta se kuitenkin tarjoaa enemmän tietoa todellisuudesta kuin arvaus tai luulo.

Erityisesti mallinnukseen ja visualisointiin tarkoitetuilla ohjelmilla pystytään jäljittelemään lähes täydellisesti realistista todellisuutta. Visualisoinnin avulla muodostunut mielikuva tarjoaa myös mahdollisuuden vaikuttaa hyvissä ajoin suunnitteluun. Lisäksi mallin avulla muutkin ihmiset kuin vain suunnittelijat pystyvät hahmottamaan kokonais kuvan. Esimerkiksi arkkitehdin uusi suunnitelma keskustasta tai asuinalueesta voidaan laittaa esille Internetiin kaiken kansan nähtäville ja kommentoivaksi. Jos suunnitelmaan ei olla tyytyväisiä, voidaan häiritsevät asiat korjata melko vaivattomasti. (Lehtovirta & Nuutinen 2000, 12–13.)

Vaikka 3D-mallintamisessa on paljon hyviä puolia, löytyy siitä myös huonoja puolia. 3D-mallien heikkoutena voidaan esimerkiksi pitää sitä, että ne eivät ole fyysisiä vaan enemmänkin pelkkiä projisoituja kuvia, joita ei ole mahdollista tunnustella, pyöritellä tai kosketella omilla käsillä. (Rajala 2012, 19) On myös olemassa asioita, jotka ei vain ole mallinnettavissa niiden suuren laskentakapasiteetin vuoksi. Tästä huolimatta 3D-mallinnus tarjoaa silti paljon enemmän mahdollisuuksia kuin esteitä ja loppuen lopuksi on vain olemassa hyvin vähän asioita joita ei voida mallintaa. Melkein voisi sanoa, että mielikuvitus on vain rajana kun kyse on 3D-mallintamisesta.

## **2.2 Kaupunkimalli**

Kaupunkimalli on kolmiulotteinen malli kaupungista. Kaupunkimallit voivat olla joko oikeista kaupungeista tehtyjä tai ne voivat olla mielikuvituksellisia. Tällaiset mielikuvitukselliset kaupunkimallit sopivat oikein hyvin esimerkiksi peleihin. Kuitenkin oikeista kaupunkimalleista on paljon enemmän hyötyä kuin kuvitteellisista, sillä ne ovat usein malleja kaupungin nykytilasta. Kolmiulotteisiin kaupunkimalleihin on mahdollista tallentaa kaikki mahdolliset hankkeet, jolloin saadaan eliminoitua päällekkäisiä töitä ja niistä syntyviä kustannuksia. (Kaupunkimalli on tulevaisuuden kantakartta 2012.)

Kaupunkimallit ovat myös erinomaisia työkaluja erilaisten analyysien ja tarkastelujen tekemiseen. Ne sopivat muun muassa erilaisten melusaasteiden tutkimiseen tai erilai-

siin tulvakartoituksiin. Mallit ovat myös helposti päivitettävissä sekä ne ovat erinomaisia havainnollistamistyökaluja. (Kaupunkimalli on tulevaisuuden kantakartta 2012; Suomisto 2013.)

Kolmiulotteisten kaupunkimallien kehitys on ollut huikea ja nopea. Ennen vuotta 2000 kaikki mallinnukset olivat manuaalisia sekä erittäin hidasta, kallista ja tehotonta. Myös tietojen käsittely ja siirto oli monimutkaista sekä koneiden piti olla todella tehokkaita. Kymmenen vuoden aikana teknologia kehittyi ja mahdollisuuksia oli huomattavasti enemmän. Muun muassa erilaiset mittausteknologiat, pistepilvet ja viisitoilmakuvat tulivat suunnittelijoiden käyttöön. Myös ensimmäinen CityGML standardi tuli käyttöön vuonna 2008. Nykyään standardien kehitys on nopeaa ja kolmiulotteiset kaupunkimallit ovat yhä useammassa projektissa mukana. (Suomisto 2013.)

### **2.2.1 Kaupunkimallin käyttötarkoitus**

Kuten aikaisemmin mainitsin, kaupunkimalleilla voidaan tehdä erilaisia analyysejä tai tutkimuksia. Esimerkiksi melukäyrien tai saaste-ennusteiden laskeminen onnistuu kun malliin saadaan liitettyä oikeat rakennuskorkeudet ja maanpinnan muodot. Tulvasimulaatioiden avulla voidaan helposti paikallistaa ongelma-alueita, jolloin suoja toimien aloittaminen voidaan aloittaa hyvissä ajoin. (Suomisto 2013.)

Kolmiulotteinen malli on myös oivallinen tapa havainnollistaa suunnitelmia muille. Mallin avulla voidaan esimerkiksi havainnollistaa kaupungin asukkaille uusia asuinalueita tai liikennereittejä. Suunnittelijoidenkin työ helpottuu kun työn voi nähdä digitaalisessa muodossa valmiina oikeassa ympäristössä. Kaupunkimalli auttaa myös hahmottamaan sitä, miten maan ja rakennusten muodot vaikuttavat valoon ja varjoihin. Tämä on erityisen tärkeää kun suunnitellaan uusia rakennuksia tai vanhoja ehostetaan.

Kaupunkimalleja on nykyään paljon tarjolla julkiseen käyttöön. Esimerkiksi Mikkeli kaupunki (2013) tarjoaa omilla verkkosivuillaan linkin jonka kautta käyttäjä voi ladata Mikkelin kaupungista kaupunkimallin (kuva 5) ilmaiseksi koneelle. (Mikkeli 2013) Myös Tampere tarjoaa verkkosivuillaan Vuoreksen kaupunginosasta hyvinkin yksityiskohtaisen mallin. Vuoreksesta tehtiin itse asiassa ensimmäinen virtuaalimalli

2000-luvun alussa ja jota on myös jälkepäin hyödynnetty kesän 2012 asuntomessu-alueella kaupunkitaiteen suunnittelua myöten. (Isotalo 2013, 17.)



**KUVA 5. Kaupunkimalli Mikkelistä**

Tukholma myös aloitti kaupungin mallintamisen vuonna 2013. Kaupunki aloitti 3D-projektin, jonka tarkoituksena on luoda uudenlainen tapa kaupungin kasvun esittämiseen ja havainnollistamiseen. Kaupunkimallia on myös tarkoitus käyttää jatkossa pohjana uusille kehitysideoille sekä tuleville suunnitelmille. Tämän vuoden aikana kaupungin olisi tarkoitus aloittaa vaihe, jossa alueiden asukkaat voivat tuoda esille omia ajatuksiaan liittyen kaupungin suunnitteluun ja alueiden käyttöön. (Tukholma kehittyi kolmiulotteisena 2013)

### 2.2.2 Kaupunkimallien tulevaisuus

Mikä sitten on kaupunkimallien tulevaisuus? Itse uskon siihen, että tulevaisuudessa kaupunkimalleja tehdään yhä useammista kaupungeista mukaan lukien Suomen kaupungeista. Se, että ne ovat vähentävät kustannuksia esimerkiksi alueiden tai tieverkostoiden suunnittelussa on tänä päivänä hyvin tärkeää. Myös CityEnginen ja Unity3D-pelimoottorin yhteistyön myötä uskon, että kaupunkimalleja tullaan näkemään yhä enemmän pelimaailmassa. Tampereen kaupungin paikkatieto- ja suunnittelupalveluiden päällikkö Ujanen sanoi hyvin Position lehden haastattelussa, että kaupunkimallit ovat tulevaisuuden kantakarttoja.

Vaikka Ujanen uskoo kolmiulotteisten kaupunkimallien olevan arkipäivää tulevaisuudessa, suosittelee hän kuitenkin kaupungin selvittämään kaikki tarvittavat tiedot ja valmiudet ennen kuin päätös kaupunkimallin rakentamisesta on päätetty. Tällä hän siis tarkoittaa sitä, että on hyvä selvittää esimerkiksi koneiden ja ohjelmien saatavuus sekä tietoa rakennusten korkeuksista jotta niiden tuomia etuja voidaan käyttää kaupunkimallissa. Näillä tiedoilla ja työkaluilla päästään hänen mukaan jo pitkälle, mutta se mitä kaikista eniten tarvitaan, on riittävä innostus, yhteistyökyky ja halua laittaa kaikki vanhat menetelmät uusiksi. (Isotalo 2013, 19.) Itse olen myös samaa mieltä, että kun nämä tiedostetaan, saadaan kaupunkimallista täydellinen hyöty irti ja erinomainen työkalu tulevaisuuden projekteille.

### **2.3 Kaupunkimallin aineistot**

Kaupungin mallintamiseen voidaan käyttää useita erilaisia aineistoja joista seuraavaksi käyn läpi muutamia muun muassa tässä työssä käytettyjä aineistoja. Yleisin näistä aineistoista on kuitenkin laserkeilausaineisto, mutta koska kyseinen aineisto ei kuulu tähän työhön, en aio siitä tämän enempää mainita. Aineistot kuten kartat ja kuva-aineistot ovat enemmän tämän työn aiheena. Ennen niitä on kuitenkin hyvä tietää termit paikkatieto ja paikkatietoaineisto. Lyhyesti sanottuna nämä kuva-aineistot ja kartat sisältävät tietoa jota kutsutaan paikkatiedoksi ja kun nämä tiedot ovat digitaalisessa muodossa, syntyy paikkatietoaineisto.

Digitaalisessa muodossa olevaa tietoa on myös saatavilla Internetistä. Esimerkiksi Maanmittauslaitos jakavat karttoja muille käyttäjille vapaasta. Maanmittauslaitoksen lisäksi on olemassa Internet-karttapalveluja. Tällaisia palveluja ovat muun muassa Google Maps ja OpenStreetMap. Nämä molemmat palvelut tarjoavat käyttäjälle mahdollisuuden tarkastella kaupunkia selaimella sekä etsiä reittivaihtoehtoja ja rakennuksia.

#### **2.3.1 Paikkatieto**

Paikkatiedolla tarkoitetaan kaikkea tietoa, jotka sisältävät viittauksen tiettyyn paikkaan tai maantieteelliseen alueeseen. Se koostuu karttatiedosta eli sijaintitiedosta sekä ominaisuustiedosta. Ominaisuustiedolla kerrotaan käyttäjälle tietoa alueesta ja tämä tieto voi olla esimerkiksi rakennuksen kerroslukumäärä, metsätyyppi tai alueen väki-

luku. Sijaintitiedon avulla saadaan sijoitettua kohde kartalle ja se voidaan ilmoittaa maantieteellisten koordinaattien tai osoitteiden lisäksi myös kohteen muoto pisteinä ja viivoina eli rasterimuotoisena aineistona tai alueina eli vektorimuotoisena aineistona. Yhdessä ominaisuustieto ja sijaintitieto muodostavat paikkatietoaineiston. (Nieminen 2011, 21; Esri Finland 2012; Vesseli 2014.)

### **2.3.2 Paikkatietoaineisto**

Paikkatietoaineistoa löytyy monesta eri paikasta ja sitä keräävät esimerkiksi valtion virastot ja yksittäiset ihmiset. Tämä aineisto on digitaalisessa muodossa olevaa aineistoa joka siis muodostuu sijaintitiedosta ja ominaisuustiedosta. Paikkatietoaineiston voi muodostaa esimerkiksi taulukko, jonne on tallennettu tietyn maantieteellisen alueen kaikkien kerrostalojen sijaintiedot sekä ominaisuustietoina jokaisen rakennuksen kerroslukumäärä sekä asukasluku. (Löytönen ym. 2003; Nieminen 2011, 21; Paikkatietoa on kaikkialla 2014.)

Paikkatietoaineistot tallennetaan tietokoneelle usein tietokannoiksi. Aineistoja voidaan taas lukea ja tarkastella erilaisilla paikkatieto-ohjelmistoilla. (Paikkatietoa on kaikkialla 2014.) Paikkatieto-ohjelmistot muodostavat aineistojen, palveluiden sekä henkilöiden kanssa paikkatietojärjestelmän (Esri Finland 2012).

Paikkatietojärjestelmiä kutsutaan englanniksi GIS, joka tulee englanninkielisistä sanoista Geographic / Geographical Information System. Alussa sanalla geo tarkoitetaan maantiedettä, jolla tarkoitetaan maapallon pinnalla tapahtuvia prosesseja, joita paikkatiedon avulla voidaan esittää. Informaation sanalla tarkoitetaan aineistojen tai tiedon käsittelyä, varastointia ja analysointia. Sanalla System taas kuvataan koko järjestelmää johon kaikki osa-alueet yhdistetään. (Löytönen ym. 2003, 12.)

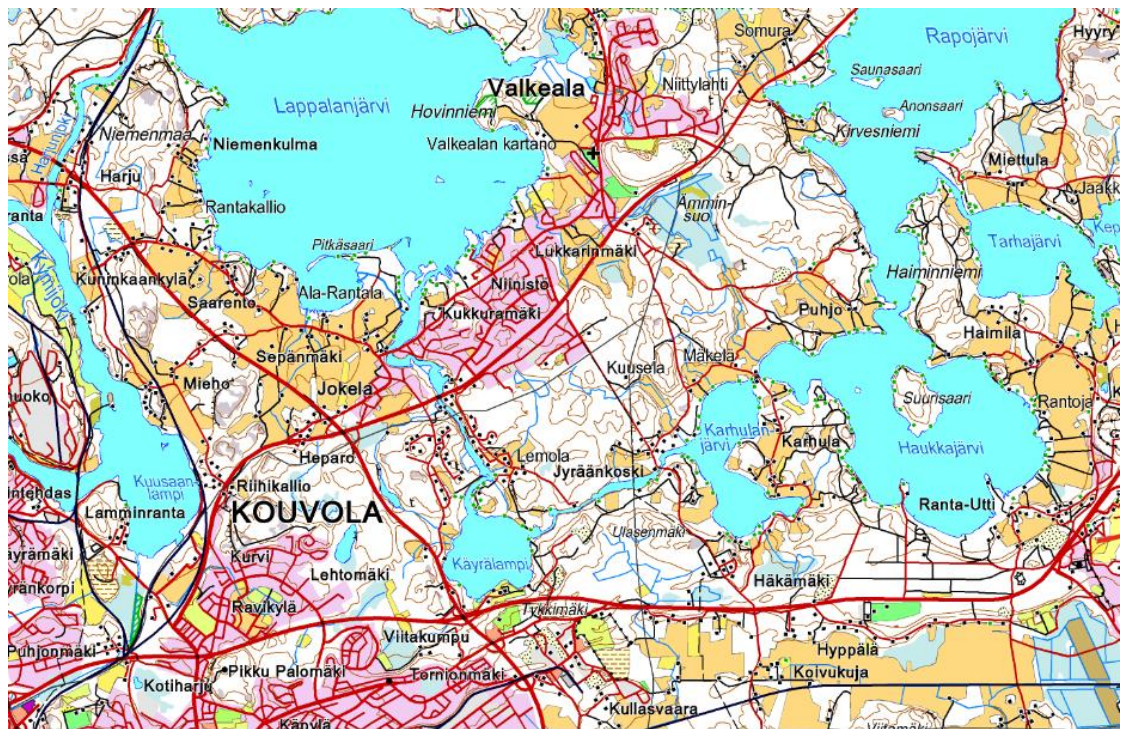
### **2.3.3 Kartat, kuva-aineistot ja Internet-karttapalvelut**

Kartta on visuaalinen esitys tietystä maantieteellisestä alueesta yleensä kaksiulotteisena. Kartan tarkoituksena on kuvata tiettyä aluetta laajasti sekä tarjoamalla samalla siitä käytetyn mittakaavan ja mahdollisimman selkeän yleiskuvan kyseisestä alueesta. Karttojen tekemiseen käytetään tieteenalaa nimeltään kartografia.

Kartografialla tarkoitetaan siis alaa joka on erikoistunut tuottamaan karttoja. Kartografia on erittäin tehokas tapa kuvata ja analysoida kaksi- tai kolmiulotteisessa tilassa esiintyviä asioita, muotoja ja sijaintisuhteita. Aiemmin kartat tehtiin käsin, mutta tekniikan kehitettyä tietokoneet ja paikkatieto-ohjelmistot ovat syrjäyttäneet vanhat menetelmät. (Löytönen ym. 2003, 20.)

## Maastokartat

Kun kartassa kuvataan maastoa ja alueita, sitä kutsutaan maastokartaksi. Maastokartat ovat usein kaksi ja puoliulotteisia ja ne perustuvat ilma- ja satelliittikuvauksiin. Ne ovat karttaesityksiä tietyn alueen topografiasta eli ne sisältävät tietoa esimerkiksi vuoren pinnanmuodosta. Kuten kuvasta 6 nähdään, maastokartat näyttävät myös korkeus- suhteet, vesistöjen sijainnit, pellot, suot sekä liikenne- ja asutusalueita. (Karttojen perusominaisuudet 2014)



**Kuva 6. Maastokartta Kouvola 1:100 000 jossa siis 1cm vastaa 1000m (sisältää Maanmittauslaitoksen Maastotietokannan 03/2014 aineistoa)**

Maastokartoissa on myös mahdollista korostaa tiettyjä asioita kartan tarpeen mukaan. Maastokarttoja voidaan suunnitella esimerkiksi veneilijöille tai autoilijoille. Veneilijöille tarkoitettut kartat sisältävät yleensä tietoa merimerkeistä ja vene- ja laivaväylien

sijainneista. Autoilijoille suunnatuissa kartoissa on yleensä esitetty tietoa liikenne-oloista. (Karttojen perusominaisuudet 2014.)

### **Kantakartta**

Kunnan tai kaupungin ylläpitämä kantakartta on suuressa mittakaavassa oleva kartta-aineisto. Kantakarttaa käytetään yleensä pohja-aineistona rakentamisella, suunnittelulle ja muille kartoille. Esitettäviä kohteita sillä ovat muun muassa rakennukset, kiinteistöt, korkeustiedot ja liikenneväylät. (Laitinen 2012, 3; Kouvola 2013.)

### **Kuva-aineistot**

Ortoilmakuvat ovat ilmakuvia muun muassa kaupungeista tai maastoista ja jonka tarkkuus perustuu pikselin kokoon maastossa. Ne soveltuvat erinomaisesti erilaisten suunnitelmien pohjaksi esimerkiksi kantakartan kanssa. Ortoilmakuvista on myös mahdollista mitata esimerkiksi kohteiden etäisyyksiä tai peltojen pinta-aloja. Ne ovat myös erinomaisia lähteitä maastotiedon keräämiseen. Kuvassa 7 on ortoilmakuva Mikkelistä joka on ladattu Maanmittauslaitoksen sivuilta. (Haggrén 2002; Maanmittauslaitos 2014.)



**KUVA 7. Ortoilmakuva Mikkelistä (sisältää Maanmittauslaitoksen Maastotietokannan 03/2014 aineistoa)**

Digitaaliset satelliittikuvat ovat suurimmiksi osaksi orto-oikaistuja, mutta jos näin ei ole, niin ne täytyy geometrisesti oikaista tunnettuun karttakoordinaatistoon. Tavallisesti käytettyjen menetelmien sijasta satelliittikuvia voidaan käyttää erilaisten digitaalisten paikkatietoaineistojen päivittämiseen. Tällaisia päivitettäviä aineistoja voivat olla esimerkiksi tieaineistot tai taajamat. Kuvien avulla voidaan myös päivitellä suurempiakin alueita kerralla ja jos käytettävissä on vielä korkearesoluutioista kuvaa, onnistuu myös rakennusten päivittäminen. (Saarentaus 2000, 8.)

Viistoilmakuvat ovat luultavimmin yleisin käytetty tapa kuvata ilmasta. Työ on edullista jos vain on hyvä ja esteetön näkymä eikä tarvitse tehdä useita ylilentoja. Viistoilmakuvat on kuvattu 45 asteen kulmasta joka mahdollistaa sen, että kuvaan saadaan mukaan myös rakennusten julkisivut ja katu ympäristöä. Kuvien tarkkuuteen vaikuttaa taas kameran lentokorkeus. Viistoilmakuvat eivät kuitenkaan sisällä koordinaatteja vaan koordinaateiksi otetaan usein kameran koordinaatit. (BlomURBEX™ integroitu... 2012; Laitinen 2012, 10; Lentokuva 2014.)

3D-malleihin voidaan myös lisätä viistoilmakuvia. Näin malleista saadaan aidomman näköisiä. Viistoilmakuvat ovat esimerkiksi kaupunkimalleissa erinomainen keino saada rakenteisiin näyttävä ja oikean näköinen tekstuuri. (BlomURBEX™ integroitu... 2012)

Julkisivukuvat ovat mahdollisimman tarkkoja kuvia esimerkiksi rakennuksen ulkoseinistä. Kuva voi olla joko piirretty kuva rakennuksesta johon on lisätty seinien pitoudet ja korkeudet tai ne voivat olla oikeita valokuvia rakennuksesta. kuvassa 8 on Mikkelin ammattikorkeakoulun rakennuksesta julkisivukuva.



**KUVA 8. Julkisivukuva Mikkelin ammattikorkeakoulun rakennuksesta**

Julkisivukuvia voidaan myös käyttää 3D-malleissa tekstuureina. Tähän tarkoitukseen kuvien olisi hyvä olla sellaisia joissa on ainoastaan rakennuksen seinä. CityEngineen on esimerkiksi mahdollista tuoda kuvassa 8 näytetty kuva ja rajata tämä niin että kuvaan jää vain rakennuksen seinä.

### **Internet-karttapalvelut**

Internetistä on saatavilla kolmea erilaista karttatyyppeä: staattisia, interaktiivisia ja animaatiokarttoja. Staattiset kartat muistuttavat paperikarttoja sillä ne ovat muuttumattomia ja esittävät vain yhden näkymän kohteesta. Toisin kuin interaktiivisia karttoja, staattisia karttoja ei voi muuttaa, panoroida, suurentaa tai pienentää. Animaatiokartat käsittelevät useita karttoja, joista jokainen kuvaa tiettyä ilmiötä tietyllä ajan hetkellä. Tällaisia animaatiokarttoja ovat muun muassa sääkartat. (Aho 2009, 7.) Seuraavaksi esittelen muutaman interaktiivisen kartan joista toinen on Googlen oma Google Maps ja toinen on avoin karttapalvelu OpenStreetMap.

Google Maps on interaktiivisen kartan lisäksi myös dynaaminen eli muuttuva järjestelmä. Käyttäjä voi kartan avulla etsiä muun muassa katukarttoja, paikallisten yritysten sijainti- ja yhteystietoja sekä reittiohjeita. Googlen on myös mahdollistanut viimeisimpien päivitysten myötä sen, että joitakin kaupunkeja ja maamerkkejä voidaan tarkastella kolmiulotteisena. Tämä kuitenkin vaatii selaimelta tuen ja että kaikki tarvittavat toiminnot ja laajennukset ovat päällä sekä järjestelmävaatimukset kunnossa. Google myös tarjoaa Earth-näkymän jonka avulla käyttäjä näkee maanpinnasta 3D-sateellittikuvaa. (Google 2014a; Google 2014b.)

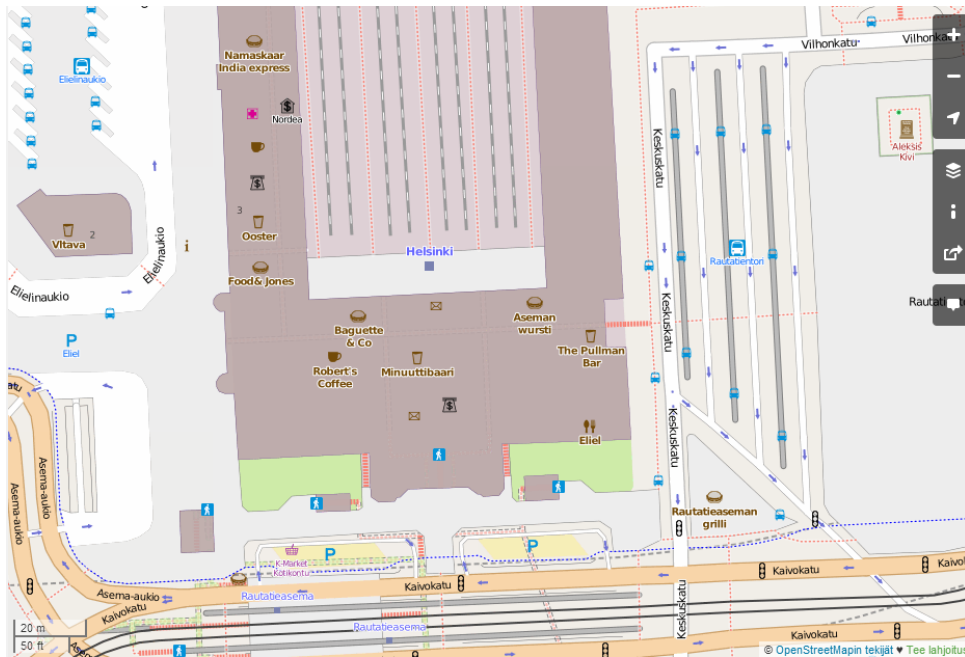
Google on myös kehittänyt vastaavanlaisen karttakehitystavan kuin OpenStreetMapilla. Google Map Makerin avulla käyttäjä voi lisäillä tai päivitellä maantieteellisiä tietoja Google Mapsiin ja Google Earthiin. (Google 2014c.) Alun perin Map maker suunniteltiin maihin, joissa Googella ei itse ollut mahdollista lisätä tietoa. Näin paikalliset pystyivät lisäämään puuttuvia tietoja ja laajentamaan Google Mapsiä. (Google Map Maker 2014)

### 2.3.4 Openstreetmap

OpenStreetMap on Steve Coastin perustuma interaktiivinen Internet-karttapalvelu (OpenStreetMap 2014). Toisin kuin monet muut kartat tai karttapalvelut, tämä maailmankartta on vapaasti ihmisten käytettävissä ja muokattavissa. OpenStreetMap pyytää ainoastaan, että käyttäjät muistavat mainita, mistä tieto on peräisin sekä kertoa tiedon olevan vapaata tietoa myös muille. Tämä palvelu on tyyliltään hyvin samanlainen kuin Googlen Maps palvelu. Erona niillä on kuitenkin se, että OpenStreetMap on avoin, eikä sitä koske samalla tavalla käyttöä rajoittavat tekijänoikeudet kuin Google Mapsia. (Keskikiikonen 2012; Sani 2009.)

OpenStreetMap tai ”avoin katukartta” on saatu muodostettua yli kymmenien tuhansien vapaaehtoisten avulla. Nämä vapaaehtoiset ovat tallentaneet omat GPS jäljet sekä mahdolliset muistiinpanot ja lisänneet ne myöhemmin palvelimelle. Kartan muodostamiseen on myös käytetty paikkojen ilmakuvia, joita on hankittu joko avoimista lähteistä tai saatu lahjoitusten myötä eri tahoilta.

Kartta on myös avoimuuden takia helppo pitää ajan tasalla. Käyttäjät voivat itse päivittää kartan heti kun huomaavat sen olevan puutteellinen. Tämän takia kartta saattaa olla hyvinkin yksityiskohtainen. Kuvassa 9 on OpenStreetMap-kartta jossa on esitetty Helsingin rautatieasema. Kuten kuvasta voi nähdä, OpenStreetMapista löytyy myös tavanomaisten teiden lisäksi rakennuksia, kahviloita ja pankkiautomaatteja. Rakennukset esitetään niiden reunaviivojen perusteella ja liikkeet sekä palvelut on merkitty erilaisilla symboleilla. Esimerkiksi Minuuttibaarin vieressä oleva musta läiskä esittää dollaria joka on pankkiautomaatin symboli.



**KUVA 9. OpenStreetMap kartta Helsingin rautatieasemasta. (© OpenStreetMapin tekijät)**

Avoimuus on myös poikunut yllättäviä käyttötapoja. Karttoja on nimittäin kohopainettu kuparilevyille näkövammaisia varten. Näkövammaisille on myös kehitetty kuparilevyjen lisäksi BlindSquare-sovelluksen, joka hyödyntää OpenStreetMapia opastuksessa. BlindSquare on suomalaisten kehittämä sovellus, joka puhuu suomen lisäksi myös ruotsia ja englantia. OpenStreetMapin ympärille on kehitetty BlindSquaren lisäksi monia muitakin avoimia ohjelmistoja. Näiden lisäksi sitä on myös käytetty erilaisten reititysten suunnitteluun sekä karttojen piirtämiseen. (Keskikiikonen 2012; BlindSquare 2014; Sani 2009.)

OpenStreetMapin avoimuus on myös sen heikkous. Tiedon luotettavuus kärsii, kun kuka tahansa voi lisätä tietoa palveluun eikä kukaan ole ottamassa vastuuta OpenStreetMapin sisältämästä tiedosta. Tämä takia palvelun käyttömahdollisuudet rajoittuvat jonkin verran. Esimerkiksi kaikkiin tärkeisiin reittiopastuksiin on hyvä käyttää niin sanotusti luotettavia karttoja sillä kukaan ei halua, että paloautot ajavat väärään paikkaan hädän tullen. (Keskikiikonen 2012.)

## 2.4 Ohjelmistot kaupunkimallinnukseen

Ohjelmistoja kaupungin mallintamiseen on monenlaisia. Ideana tällaisilla ohjelmilla on se, että niillä voidaan luoda massiivisia ja mahdollisimman tarkkoja kaupunkimal-

leja jo olemassa olevasta paikkatietodatasta. Tällaisia ohjelmia ovat muun muassa Esrin CityEngine, ArcGIS 3D Analyst, Novapointin Virtual map sekä Autodesk Infrastructure Modeler. Seuraavaksi esittelen näistä ohjelmista tarkemmin Esrin CityEnginen sekä 3D Analystin. Nämä muut ohjelmat käyn hieman suppeammin läpi. Lopuksi esittelen vielä muutamia enemmänkin arkkitehdeille tarkoitettuja ohjelmia.

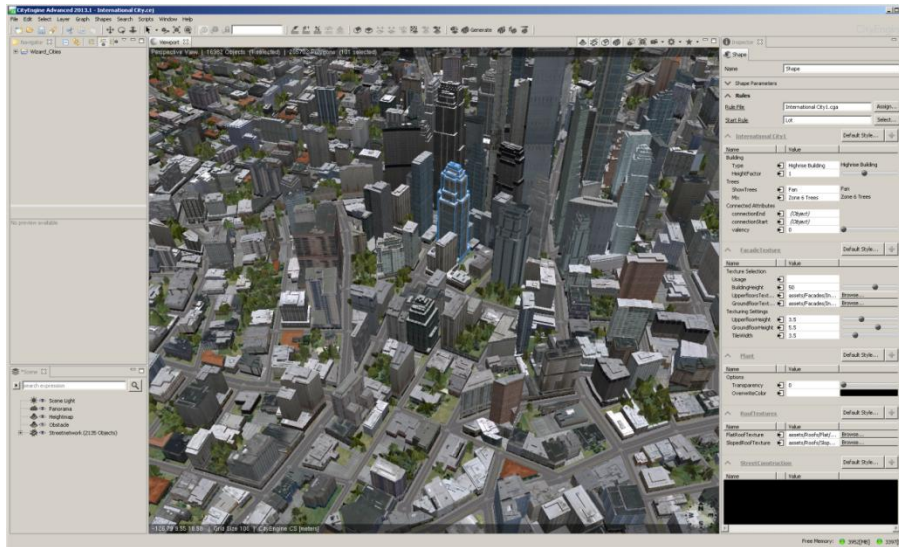
### **2.4.1 Esrin CityEngine**

Esrin CityEngine on ohjelmisto, jolla voidaan luoda tarvittaessa nopeasti kolmiulotteisia kaupunkimalleja. WebGL-tekniikan ansiosta näkymät voidaan myös esittää selaimella ilman erillisiä liitännäisosia. Esri, joka on siis maailman johtava paikkatietoratkaisujen toimittaja, osti CityEnginen vuonna 2012 sveitsiläiseltä Procedural-yhtiöltä.

Ohjelmiston avulla pystytään hyödyntämään jo olemassa olevia 2D-paikkatietoaineistoja ja muuttamaan ne erilaisten sääntöjen avulla 3D-malleiksi. CityEngine tarjoaa valmiiksi melko kattavan määrän erilaisia sääntöjä, mutta näitä sääntöjä on myös mahdollista kirjoittaa itse. CityEnginessä CGA (Computer Generated Architecture) sääntöjä voidaan luoda esimerkiksi kirjoittamalla niitä ohjelman omaan editoriin. Editoriin kirjoitetaan CGA shape grammar -ohjelmointikielellä, joka on nimenomaan tarkoitettu kolmiulotteiseen arkkitehtuuriseen mallintamiseen. CityEngine tarjoaa myös konsolin johon voidaan kirjoittaa python-ohjelmointikielellä. Kielen avulla on muun muassa mahdollista automatisoida tehtäviä. Tällaiset tehtävät voivat olla esimerkiksi kaikkien näkymässä olevien mallien muuttaminen yhteen tiedostomuotoon. (Esri tarjoaa... 2011, 20–21; Esri CityEngine 2014.)

### **Sääntöpohjainen mallinnus**

Sääntöihin perustuva mallinnus tarjoaa mahdollisuuden mallintaa kaupunkimalleja nopeasti. Sääntöillä voidaan myös luoda massiivisia 3D-malleja, jotka noudattavat annettuja kaupunkisuunnittelun säädöksiä ja ohjeita. Vaikka mallit perustuvat sääntöihin, voidaan niitä vielä myöhemmin muuttaa ja muokata. Kuvassa 10 nähdään CityEnginen käyttöliittymä sekä ohjelman omalla säännöllä tehty kaupunkimalli. (Esri tarjoaa... 2011, 20–21; Esri CityEngine 2014.)



**KUVA 10. CityEnginen käyttöliittymä sekä kuvitteellinen kaupunkimalli**

Säännöissä voidaan myös käyttää pohjakartassa olevia ominaisuustietoja hyväksi. Esimerkiksi jos pohjakarttaan on lisätty tietoa rakennuksen korkeudesta, sääntöön voidaan lisätä attribuutti joka käyttää tätä tietoa rakennuksen korkeuden määrittämiseen. CityEnginettä voidaan myös tuoda muilla ohjelmilla rakennettuja 3D-malleja. Ohjelma nimittäin tukee muun muassa Collada, Autodesk ja FBX 3D-formaatteja. CityEnginellä on myös mahdollista rakentaa katuja erikseen sekä suunnitella rakennuksia suoraan 3D:nä. (Esri tarjoaa... 2011, 20–21; Esri CityEngine 2014.)

Sääntöjen vuoksi CityEnginellä 3D-työskentely on tehokasta sekä halpaa ja on siksi erinomainen työkalu yrityksille, jotka tekevät erilaisia kaupunkimalleja. Tätä kyseistä ohjelmaa käyttävät maailmalla kaupunkisuunnittelijoiden lisäksi myös muun muassa puolustusvoimat sekä elokuva- ja peliteollisuus, joista ainakin Disney ja Blizzard ovat mainitsemisen arvoisia yhtiöitä. (Esri tarjoaa... 2011, 20–21.)

Sääntöjen lisäksi CityEngine tukee myös tyyliä. Näiden tyylien avulla on mahdollista tuottaa erilaisia rakennuksia helposti ja nopeasti. Esimerkiksi rakennusten julkisivumateriaalia, katon muotoa ja monia muita ominaisuuksia on mahdollista vaihdella hetkessä. Nämä muutokset vaihtuvat saman tien eikä rakennusta tarvitse luoda uudelleen. (Esri tarjoaa... 2011, 20–21.)

CityEngine sopii täydellisesti Esrin tuoteperheeseen, sillä se täydentää mainiosti ArcGIS 3D Analyst -laajennusosaa. Kyseinen ohjelma on tyyliltään hieman samanlainen kuin CityEngine. Ohjelmalla voidaan nimittäin samalla lailla visualisoida ja jakaa 3D-

paikkatietoa. Erona CityEngineen on se, että monipuoliset tekstuuirakennukset jäävät vielä muiden ohjelmistojen mallintamiseksi. Esrin mielestä CityEngine sopii jopa niin hyvin tuoteperheeseen, että he ovat päättäneet integroida ohjelman ArcGIS-alustaan. (Esri tarjoaa... 2011, 20–21.)

#### **2.4.2 ArcGIS ja 3D Analyst**

ArcGIS on avoin järjestelmä joka koostuu paikkatieto-ohjelmistoista, paikkatietokannoista ja paikkatietopalvelimista. Tämän lisäksi se on myös paikkatietojärjestelmä. ArcGIS:lla on siis mahdollista katsella, luoda, visualisoida, hallita, analysoida ja jakaa paikkatietoa. Tähän kyseiseen järjestelmään pääsee käsiksi erilaisilla välineillä kuten esimerkiksi tietokoneisiin asennetuilla ArcGIS-ohjelmistoilla tai mobiililaitteilla. Avoimen järjestelmän vuoksi sovelluskehittäjät voivat myös kehittää helposti ja nopeasti omia selain-, työasema- tai mobiilisovelluksia. (ArcGIS on kokonaisjärjestelmä 2011, 15.)

ArcGIS aineistot ovat mahdollista tallentaa esimerkiksi muistitikuille tai ne voivat sijaita organisaation sisäverkossa. Kyseisen järjestelmän on myös mahdollista hyödyntää pilvipalveluita tai vaikka pyöriä ainoastaan pilvessä. Tärkeintä kuitenkin on, että kaikki järjestelmän osat ovat yhteydessä toisiinsa. Tämä myös tarkoittaa sitä, että jos tietokannassa muutetaan jotain, se näkyy samanlaisena jokaisessa päätelaitteessa ja sovelluksessa. Kuitenkin kaikilla eri laitteilla voidaan myös luoda ja tallentaa paikkatietoa eri tietolähteisiin. Kokonaisuena järjestelmänä ArcGIS siis tarjoaa käyttäjille mahdollisuuden hyödyntää paikkatietoa ensin johonkin rajattuun ajankohtaiseen tarpeeseen ja laajentaa järjestelmää tarpeiden kasvaessa. (ArcGIS on kokonaisjärjestelmä 2011, 15.)

#### **ArcGIS 3D ANALYST**

ArcGIS 3D Analyst on laajennusosa, joka on tarkoitettu ArcGIS for Desktop Basiciin. Laajennuksella voidaan luoda, visualisoida ja analysoida kolmiulotteisia paikkatietoja. Sen avulla on mahdollista luoda esimerkiksi realistinen simulaatio kaupungista, maisemasta, projektista tai kriittisestä tilanteesta. Erityisesti kaupunkisuunnittelussa 3D Analyst on kätevä, sillä se auttaa visualisoimaan hankkeiden vaikutuksia ja mahdollistaa tiedon jakamisen muille. Ohjelma tarjoaa myös monelle muulle oivallisen työka-

lun. Esimerkiksi arkkitehdeille se tarjoaa tavan havainnollistaa asiakkaalle kolmiulotteisia kuvia, tieliikenneyhtiöille oivan työkalun verkostojen suunnitteluun sekä poliisille työkalun jolla on mahdollista saada parempia tilannekuvia operaatioista. (3D-visualisoinnille paljon käyttökohteita 2011, 33; ArcGIS 10 monipuolistaa... 2011, 33.)

3D Analystilla on mahdollista tuottaa kolmiulotteisia malleja sekä pyöritellä ja käsitellä niitä suoraan näytöllä. Käsittelemisellä tarkoitan sitä, että niitä on esimerkiksi mahdollista siirtää ja skaalata erikokoisiksi tai jopa korvata muilla malleilla. Ohjelman esitysgrafiikka on myös erinomainen, sillä ohjelma niputtaa pikseleitä yhteen kaikkien pikselisolujen esittämisen sijasta. Näin ollen lähimpänä olevat kohteet piirtyvät tarkasti ja kauempana olevat kohteet epätarkemmin. 3D Analyst tarjoaa myös käyttäjille useita erilaisia 3D-symboleja. Tällaisia symboleja ovat muun muassa valmiit rakennukset, autot, puut, katuvalot ja muut maisemaelementit. (3D-visualisoinnille paljon käyttökohteita 2011, 33; ArcGIS 10 monipuolistaa... 2011, 33.)

### **3D Analystin analyysit**

3D Analystilla on myös mahdollista tehdä erilaisia analyysejä. Esimerkiksi varjoanalyysin avulla voidaan tarkastella mikä on uuden rakennuksen varjoalue eri kellonaikoihin ja mitkä kaikki jäävät tämän varjoalueen sisälle. Ohjelman tarjoaa myös analysointityökaluja 3D-kappaleille. Näillä työkaluilla voidaan esimerkiksi tutkia puistovalojen katve-alueita tai räjähdysten vaikutusalueita. Käyttäjälle on myös annettu mahdollisuus tutkia kaupunkia erilaisista näkökulmista. (3D-visualisoinnille paljon käyttökohteita 2011, 33; ArcGIS 10 monipuolistaa... 2011, 33.)

### **2.4.3 Muita kaupunkimallinnusohjelmistoja**

Novapoint Virtual Map on ohjelma ja erityisesti työkalu, joka on tarkoitettu erilaisten rakennushankkeiden suunnitteluun, mallintamiseen ja visualisointiin virtuaalisesti. Ohjelmalla on myös tarjolla sketch-toiminto, joka mahdollistaa nopean mallintamisen. Tämän työkalun avulla käyttäjä voi piirtää 2D muotoja, mutta havainnollistaa ne 3D muodossa. Virtual Mapissa suunnittelijat voivat jakaa myös omia mallejaan keskenään XREF-linkin välityksellä. Linkin hyöty on projektien kannalta erinomainen, sillä mallintaminen ja suunnittelu tapahtuvat yhtäaikaaisesti jolloin itse projekti on myös halvempi. (Novapoint Virtual Map 2014.)

Autodesk Infrastructure Modeler on hyvin samanlainen kuin muutkin kaupunkimallinnukseen tarkoitettut ohjelmisto. Ohjelmaa voidaan muun muassa käyttää kokonaisien kaupunkien mallintamisen sijaa myös korttelien tai kanavien mallintamiseen. Lisäksi Infrastructure Modeler on yhteensopiva muidenkin Autodeskin tuotteiden kanssa. (Visualisoi menestys... 2013.)

### **Arkkitehtuuriohjelmistot**

Markkinoilla on myös olemassa ohjelmistoja joilla saadaan kaupunkimalleja arkkitehtuurin näkökulmasta. Tällaiset ohjelmisto ovat enemmänkin tarkoitettu rakennusten, pihamaiden tai puistojen suunnitteluun. Kuitenkin tällaiset ohjelmistot kuten SketchUp, ArchiCAD, Autodeskin Revit ja Vectorworks Architect ovat myös sopivia kaupunkien mallintamiseen.

SketchUp ei varsinaisesti ole arkkitehtuuriohjelmisto tai kaupunkimallinnusohjelma, mutta se sitä voidaan käyttää molempiin tarkoituksiin. Ohjelma sopii erinomaisesti erilaisten tilojen ja tuotteiden suunnitteluun sekä havainnollistamiseen. SketchUpia voidaan myös käyttää rakennusten sekä kaupunkialueiden mallintamiseen. Tästä on hyvänä esimerkkinä aikaisemmin mainittu Mikkelin kaupunki jonka keskustan 3D-malli löytyy kaupungin sivuilta. Itselleni ohjelmasta tulee mieleen Microsoftin Paint-ohjelma, mutta vain huomattavasti parempi ohjelma.

ArchiCAD on työkalu joka tarkoitettu erityisesti suunnittelijoille, sillä se on kehitetty rakennussuunnittelijan näkökulmasta. Ohjelma tarjoaa käyttäjälle tai asiakkaalle pohjapiirustusten lisäksi myös muiden ohjelmien tavoin kolmiulotteisen kuvan rakennuksesta. ArchCAD antaa myös mahdollisuuden esittää tämän rakennuksen tai tilan animaation avulla. (Micro Aided Design 2014.)

Autodesk Revit on hyvin samanlainen kuin ArchCAD. Ohjelmisto on tarkoitettu rakennusten suunnitteluun sekä rakentamiseen. ”Revit on kehitetty erityisesti tietomallinnukseen (BIM, building information modeling), ja siinä on kattavat ominaisuudet, joiden ansiosta se on ihanteellinen ratkaisu koko rakennustiimille.” (Autodesk Revit 2014) Kyseinen ohjelmisto omaa muun muassa arkkitehtisuunnittelua ja rakenteiden rakentamista varten erinäisiä ominaisuuksia.

Vectorworks Architect on CAD-ohjelma jossa onnistuu kaksi ja kolmiulotteinen suunnittelu samanaikaisesti. Ohjelma on tarkoitettu muun muassa arkkitehtuurisuunnitteluun, ympäristösuunnitteluun sekä rakennelmien ja rakennusosien suunnitteluun. Architect sisältää valmiiden symbolien lisäksi vielä CAD details kirjastoja sekä näiden lisäksi vielä erilaisia työkaluja esimerkiksi rakennusten massoitteeluun. Ohjelmaan voidaan tuoda myös GIS-aineistoja. (Ultirender 2014.)

### **3 KAMPUKSEN MALLINTAMINEN CITYENGINELLÄ**

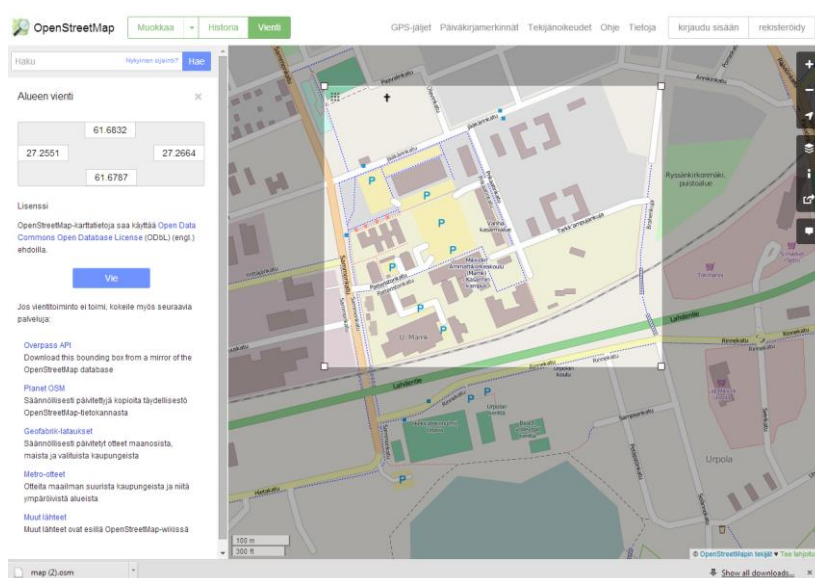
Tämän työn tarkoituksena on selvittää, kuinka kaupunkimallinnusohjelma CityEngine soveltuu Mikkelin ammattikorkeakoulun kampuksen mallintamiseen sekä minkälaisia erilaisia menetelmiä ohjelma tarjoaa rakennusten mallintamiseen. Työssä perehdytään myös OpenStreetMapista ladattavaan OSM tiedostoon ja kuinka tätä tiedostoa voidaan käyttää ohjelmassa. Viimeisenä vielä selvitetään, kuinka malli voidaan viedä selaimelle. Tämä työn on osa koulun paikkatietojärjestelmän ja 3D-mallinnusopetuksen kehitystä

Tämä työ on jaettu kahteen osaan, jossa ensimmäisessä osassa on tarkoitus tutkia sekä kokeilla, kuinka OpenStreetMap-tiedosto voidaan tuoda ohjelmaan. Ensimmäisessä osassa kokeillaan myös CityEngine tarjoamia erilaisia mallinnusmenetelmiä tontteihin sekä katuverkostoon. Toisessa osassa perehdytään, kuinka malli voidaan viedä selaimelle käyttäen ohjelman omaa Web Vieweriä. Ohjelmalla työskentäminen on siinä mielessä hankalaa, sillä en ollut tietoinen ohjelmasta ennen kuin aloin työskentämään tätä opinnäytetyötä.

Aloitin tämän työn työskentämisen lataamalla aluksi OSM tiedoston OpenStreetMap-sivustolta. Tämän jälkeen kokeilin, kuinka ohjelman omia sääntöjä voidaan käyttää OSM tiedoston avulla luotuun katuverkkoon. Katuverkon jälkeen kokeilin, kuinka säännön kirjoittaminen tapahtuu. Kirjoitin kaksi erilaista sääntöä, joista toisella säännöllä määrätään luomaan rakennus tekstuureineen ja toisella määrätään rakennuksen korkeus tontissa määrättyllä attribuutilla. Tämän jälkeen kokeilin, kuinka sääntöjä voidaan luoda kuvien perusteella käyttäen ohjelman facade wizard -toimintoa. Ennen mallien viemistä selaimelle, kokeilin vielä kuinka rakennuksen mallintaminen onnistuu suoraa tontille käyttäen ohjelman polygon- ja rectangular shape creation -työkaluilla.

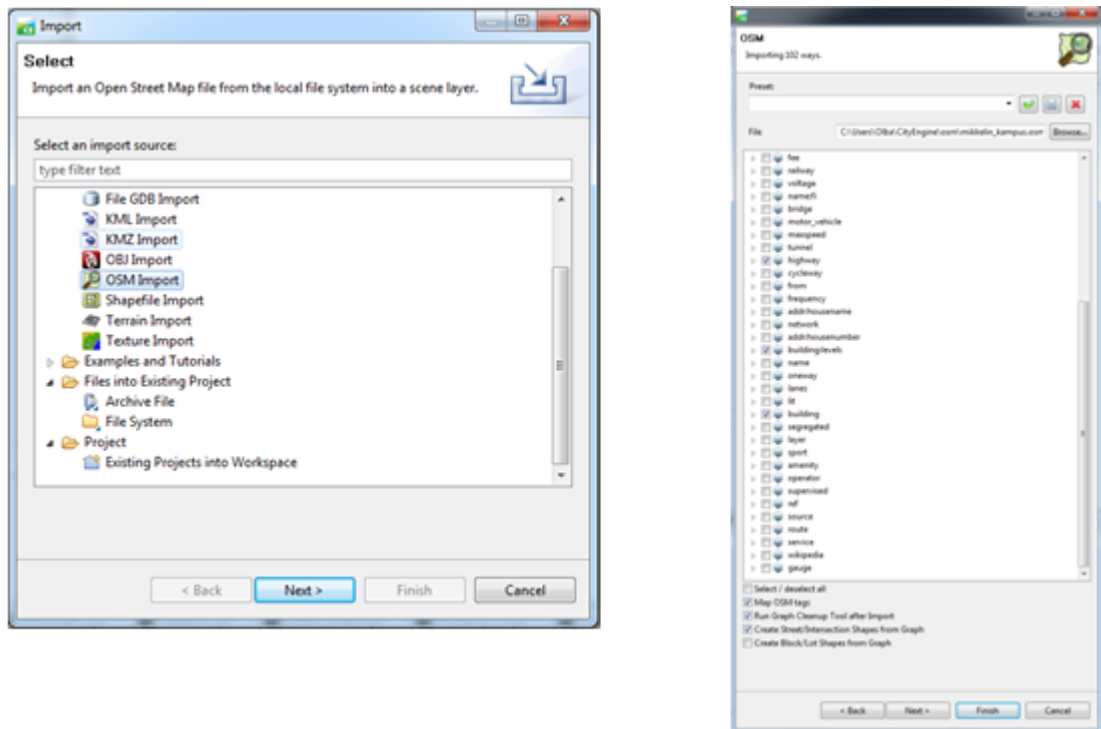
### 3.1 OpenStreetMap & CityEngine

Työn ensimmäisenä vaiheena oli ladata OpenStreetMap-sivulta Mikkelin ammattikorkeakoulun alueesta OSM tiedosto. OSM on XML-pohjainen tiedosto vektoriaineistojen kuvaamiseen. Tiedostossa kerrotaan elementtien sisällä muun muassa jalkakäytävän tai rakennuksen muoto. Elementtien sisälle on voitu myös lisätä tietoa alueesta. Tällainen tieto voi olla esimerkiksi jalkakäytävän nimi tai rakennuksen korkeus. CityEngine käyttää näissä elementeissä kerrottuja tietoja pohjakartan muodostamiseen. Ohjelmaan ei kuitenkaan ole pakko tuoda kaikkia elementtejä vaan halutut elementit voidaan valita siinä vaiheessa kun OSM-tiedostoa tuodaan näkymään import-toiminnolla. Kuvassa 11 on esitetty OpenStreetMap sivu sekä alueen rajaaminen.



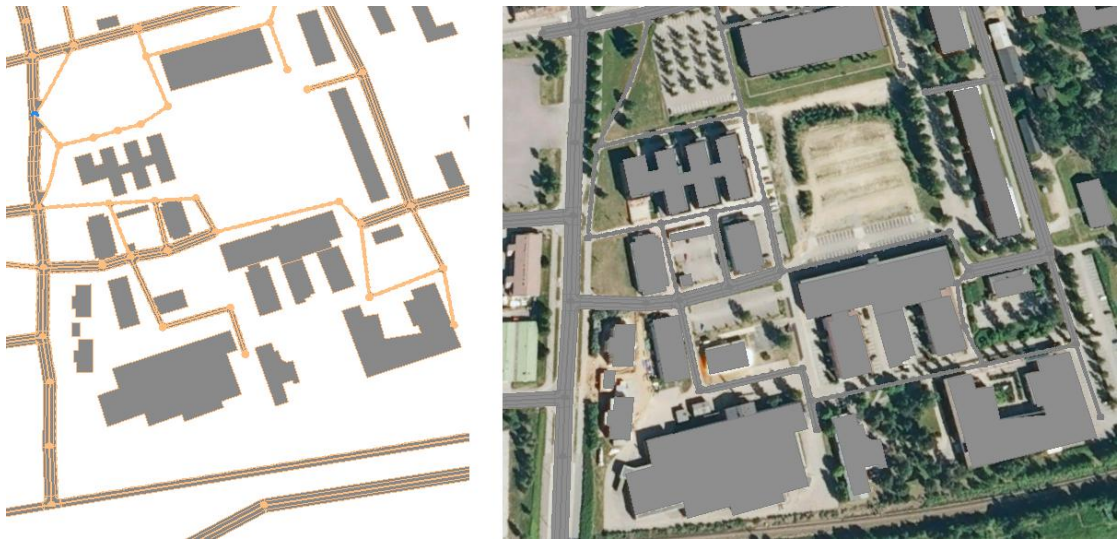
**KUVA 11. Kuvassa on OpenStreetMap ja alueeksi on rajattu Mikkelin ammattikorkeakoulu alue**

Kuten mainitsin, OSM tiedosto voidaan tuoda CityEnginen näkymään ohjelman import-toiminnolla. Import-valikossa määritellään muun muassa, mitkä kaikki elementit halutaan tuoda näkymään sekä halutaanko vektoriaineistoa korjata ohjelman omilla toimintoilla. Tähän työhön valitsin ainoastaan tiet ja rakennus-elementit sekä katuverkon korjaamisen. Korjaaminen tapahtuu vähentämällä ylimääräisiä pisteitä ja yhdistelemällä joitakin pisteitä toisiinsa. Elementtejä on myös mahdollista tuoda erillisille tasolle tuomalla OSM-tiedosto uudestaan näkymään. Kuvassa 12 on esitetty import-valikko sekä oikealla OSM-tiedoston sisältämiä elementtejä.



**KUVA 12. Vasemmalle import-ikkuna ja oikealla elementtien lista**

Katuverkostoa on myös mahdollista korjailla vielä näkymässä joko cleanup graph -työkalulla tai käsin. Tämän katuverkoston korjaaminen tapahtui käsin, sillä omasta mielestä pienemmän alueen korjaaminen onnistuu parhaiten näin. Korjasin katuverkkoa muun muassa yhdistämällä joitakin katuja toisiinsa sekä rakentamalla uusia ohjelman freehand- ja polygonal street creation -työkaluilla. Kuvassa 13 näemme vasemmalla OSM-katuverkon heti import-toiminnon jälkeen ja oikealla näkymän, jossa katuverkko on korjattu sekä katuverkon alapuolelle on tiputettu alueesta otettu ortokuva. Ortokuvan asetin alueeseen siirtämällä tiedoston näkymän päälle. Kuva oli liian suuri katuverkostoon nähden, joten jouduin skaalaamaan kuvan pienemmäksi ohjelman skaalaus-työkalulla. Tämän jälkeen asettelin vielä joitakin katuja ja rakennuksia ortokuvan mukaisesti.



**KUVA 13. Vasemmalle OSM tiedosto importattu ja oikealla OSM tiedostoa korjailtu ja alle on liitetty ortokuva**

### 3.1.1 Sääntöihin perustuva mallinnus

CityEnginessä mallinnus perustuu pääsääntöisesti sääntöihin. Näillä säännöillä määrätään esimerkiksi rakennuksen muoto sekä tekstuurit. CityEngine tarjoaa erilaisten harjoitusten ja esimerkkien myötä erilaisia sääntöjä, mutta näitä sääntöjä voidaan myös kirjoittaa itse ohjelman editorissa tai työkaluilla.

CityEnginen CGA sääntöjä kirjoitetaan CGA shape grammar -ohjelmointikielellä. Tämä kieli on nimenomaan suunniteltu luomaan kolmeulotteisia malleja CityEnginenäkymään. Sääntöpohjaisessa mallinnuksessa ideana on luoda sääntöjä, joissa malleja kehitetään toistuvasti luomalla niille uusia yksityiskohtia. CGA-sääntö muodostuu muodoista, joita kutsutaan muotosymboleiksi. Tämä taas koostuu parametreistä, attribuuteista, geometrioista, näkyvyysalueesta (scope) sekä pivotista. Muodon geometria on mesh-verkko. Tämän lisäksi geometrioihin luetaan muun muassa materiaalit ja tekstuurit.

Perusidea sääntöä kirjoittaessa on korvata muoto uudella muodolla, jolle on määritelty liuta uusia muotoja. Kuvassa 14 on esitetty tekstieditori sekä havainnollistettu, kuinka säännössä ensimmäinen muoto (tontti) korvataan uudella muodolla joka on nimeltään rakennus. Ensimmäisessä muodossa määrättyjä toimintoja ei tietenkään hylätä vaan ne kopioidaan ja siirretään seuraavalle muodolle joka on tässä tapauksessa rakennus. Valmista rakennetta kutsutaan puurakenteeksi jossa ensimmäistä muotoa kutsutaan

juureksi ja siitä lähteviä muotoja lehdiksi. Lehdet muodostavat siis loppuen lopuksi mallin.

Kuvassa 14, juuressa extrude-operaattorilla nostetaan tontti attribuutissa määrätylle korkeudelle. Juuressa myös luodaan uusi lehti nimeltä rakennus, jossa operaattorilla comp() jaetaan rakennus osiin. Rakennus saadaan jaettua osiin antamalla sille parametrin f (face). Tämä luo lehdelle rakennus neljä ns. lasta, joille voidaan määrittellä erilaisia operaattoreja joita rakennuksen tulee noudattaa.



```

attr korkeus          = 3.5

tontti-->
  //nostetaan seinät 3.5m korkeuteen
  extrude(korkeus) rakennus

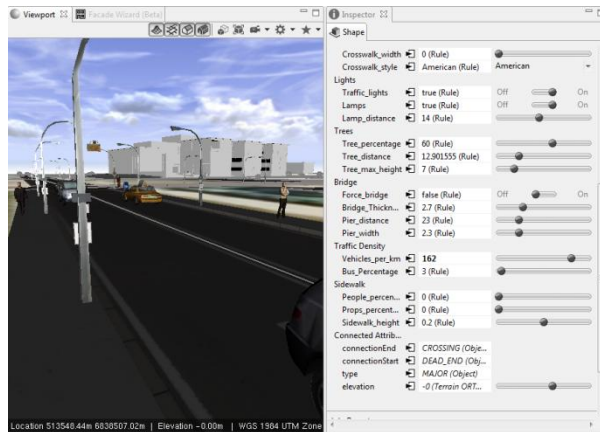
rakennus-->
  //määritellään rakennukselle sivut ja katto
  comp(f) {front : etuseina | back : takaseina | side : sivut | top : katto}

```

#### KUVA 14. CGA tekstieditori sekä säännön idean havainnollistaminen

Ohjelman harjoitusten ja esimerkkien sääntöjä voidaan myös käyttää omissa töissä. Näiden sääntöjen määrääminen onnistuu suoraan toisista projekteista, mutta ne voidaan myös halutessa kopioida omaan projektiin. Ennen säännön kopioimista täytyy kuitenkin muistaa säännössä käytettyjen tekstuurien sekä objektien kopioiminen omaan projektiin. Tämä siitä syystä, että säännöissä kirjoitettujen tekstuurien ja objektien polku toimisi. Jos tekstuurit tai objektit eivät toimi, tarvitsee säännön polku kirjoittaa uudelleen.

Työssä kokeilin kuinka ohjelman modern street esimerkkiä voidaan käyttää omissa projektissa katuverkkojen mallintamiseen. Sääntö ja sekä säännössä käytetyt tekstuurit ja objektit ovat kopioitu oman projektin kansioihin. Säännössä määritellään tielle ja jalkakäytävälle omat muodot, jotka mukautuvat niiden leveyksien mukaan. Esimerkiksi leveämmälle jalkakäytävälle voidaan tuoda attribuuttien avulla enemmän objekteja kuin kapeammalle. Attribuutteja muuttelemalla voidaan määrätä leveälle jalkakäytävälle muun muassa ihmisiä, puita ja lyhtypylväitä. Tielle on mahdollista lisätä autojen lisäksi myös busseja ja suuntaviivoja. Kuvassa 15 on esitetty osa katuverkosta säännön asettamisen jälkeen sekä säännössä määritetyistä attribuuteista.



**KUVA 15. Ohjelman Modern Street esimerkin esittäminen**

Osa tätä työtä oli kirjoittaa kaksi erilaista sääntöä, joista ensimmäinen sääntö luo kattoineen ja tekstuureineen mallin sekä sisältää erilaisia attribuutteja, joiden avulla käyttäjä voi muuttaa rakennuksen ulkonäkyä. Toinen sääntö on tarkoitettu luomaan yksinkertaisia malleja käyttäen rakennuksen korkeuden määrittämiseen tontissa tarjottua korkeus attribuuttia. Korkeus attribuutti piti lisätä itse, sillä tonteista sitä ei löytynyt. Kuten olen aikaisemmin maininnut, OpenStreetMap on avoin karttapalvelu ja näin ollen OSM tiedoston sisältö riippuu siitä, mitä alueen tekijät ovat lisänneet. Jotkin alueet saattavat sisältää tietoa esimerkiksi rakennuksien korkeuksista ja toiset alueet taas ainoastaan rakennuksien nimet. Tämän lisäämisen tarkoituksena oli kokeilla kuinka tällaisia attribuutteja voidaan käyttää jos sellaisia on määriteltä.

Seuraavaksi esittelen säännön, jota käytin useamman rakennuksen mallintamiseen. Kuten mainitsin, niin rakennuksen korkeuden määrittämiseksi säännössä käytetään tonttiin lisättyä korkeus attribuuttia. Koodia ei tarvitse paljoa kirjoittaa, sillä rakennuksen muodostamiseen riittää korkeus-attribuutti sekä extrude-operaattori. Jotta tontin attribuuttia voidaan käyttää säännössä, tarvitsee attribuutin nimi kirjoittaa sääntöön sekä asettaa sille arvoksi nolla. Tämä sääntö on esitetty kuvassa 16 yhdessä rakennuksien kanssa jotka noudattavat tätä sääntöä.

```

@Range ("LOD1", "LOD2")
attr LOD = "LOD1"

attr korkeus = 0

Lot-->
  extrude(korkeus) rakennus

rakennus-->
  comp(f) {front : etuseina |
          side : sivut | top : katto}

katto-->
case LOD == "LOD1" : LOD1
case LOD == "LOD2" : LOD2
else : tasokatto

LOD1-->
tasokatto
color(1,0,0)

LOD2-->
case kattotyyl_i_LOD2 == "hip" : hipKatto
case kattotyyl_i_LOD2 == "gable" : gableKatto

```



**KUVA 16. Vasemmalla CGA sääntöä ja oikealla rakennukset LOD1 tasossa**

Toisessa säännössä rakennukselle lisätään seinien lisäksi ikkunoita, katto sekä muutamia erilaisia tekstuureja. Kuvassa 17 on esitetty, kuinka rakennuksen seinät jaetaan comp(f)-operaattorilla neljään eri osaan. Tämän tein siitä syystä, että voin luoda jokaiselle seinälle erilaisen pinnan. Jaoin vielä etuseinän y-suunnassa kahteen osaan split-operaattorilla. Tämän tarkoituksena on havainnollistaa, kuinka rakennukselle voidaan lisätä kerroksia rakennuksen korkeuden muuttuessa suuremmaksi. Kuvasta nähdään myös kuinka alin kerros jaetaan x suunnassa split-operaattorilla niin, että seinälle saadaan ikkunoiden paikat sekä ovelle oma kohta ikkunoiden välistä. Samaisessa kuvassa on esitetty kuinka seinälle voidaan lisätä tekstuuri. Tekstuurien asettamisessa texture komennolla saadaan valittua tekstuurille pinta ja setupProjection avulla määrätään kuinka tekstuuri projisoidaan pinnalle. ProjectUV avulla tekstuuri asetetaan paikoilleen.

```

rakennus-->
//määritellään rakennukselle sivut ja katto
comp(f) {front : etuseina | back : takaseina
        | side : sivut | top : katto}

etuseina-->
split(y) {alinkerros_korkeus : alinKerros |
        {-muut_kerrokset : kerrokset}*}

alinKerros-->
//jaetaan ensimmäinen kerros reunoihin,
//ikkunaväleihin, ovikohtaan
split(x)
{~1 : seinä |
{-ikkunoiden_vali : vali}* |
3 : oviKohta |
{-ikkunoiden_vali : vali}* |
~1 : seinä}

seinä-->
setupProjection(0,scope.xy,10,10)
projectUV(0)
texture("images/kuvia/tiiliseina.JPG")

kerrokset-->
//jaetaan muut kerrokset reunoihin ja ikkunaväleihin
split(x){~1 : seinä | {-4 : vali}* | ~1 : seinä}

```



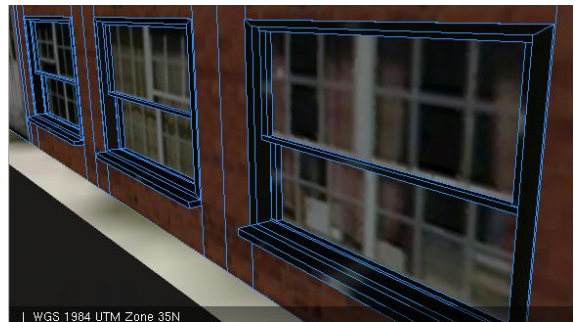
**KUVA 17. Rakennuksen seinien jakaminen sekä teksturoiminen**

Sääntö sisältää myös ehtolauseen jossa käyttäjä voi valita attribuutilla LOD2 tai LOD3 tarkkuustason. Tasossa kaksi rakennukselle on asetettu ainoastaan tekstuurit ja katto. Kolmannessa tarkkuustasossa rakennukseen tuodaan ikkunaobjekti ikkunatekstuurin tilalle. Tämä objekti on kopioitu CityEnginen basic shape grammar-harjoituksesta omaan projektiin. Jotta objekti sopisi ikkunan paikalle, täytyy sille määrittellä muutama operaattori. Operaattorilla s muutetaan ikkuna pinnan kokoa objektille sopivaksi ja operaattorilla t siirretään ikkunan pinta rakennuksen seinästä syvemmälle. Operaattorilla i tuodaan objekti sääntöön ja tutulla texture-komennolla määritellään objektille tekstuuri. Kuvassa 18 on esitetty ehtolause sekä kuinka ikkunaobjekti tuodaan ikkunan paikalle. Näiden lisäksi kuvassa on esitetty myös funktio fileRandom jolla voidaan määrätä tekstuuri satunnaisesti käyttämällä \* etuliitettä.

```
//määritellään attribuutti jonka arvo voi olla joko LOD2 tai LOD3
@Range ("LOD2", "LOD3")
attr LOD = "LOD2"

//valitaan ikkunoille satunnaisesti tekstuuri kansiota textures
ikkunan_tekstuuri = fileRandom("**facades/textures/window.*.tif")

ikkuna-->
//jos LOD3 on valittuna, vaihdetaan ikkunatekstuuri ikkunaobjektiksi
case LOD == "LOD3" :
  s ('1','1,0.4)
  t (0,0,-0.25)
  i ("facades/window.obj")
  texture(ikkunan_tekstuuri)
else :
  //lisätään ikkunan kohtaan ikkunatekstuuri
  setupProjection(0, scope.xy, scope.sx, scope.sy)
  texture(ikkunan_tekstuuri)
  projectUV(0)
```



**KUVA 18. Objektin lisääminen sääntöön**

CityEngine sisältää valmiiksi joitakin katto-operaattoreja. Yhden tällaisen operaattorin käyttämisestä on havainnollistettu kuvassa 19. Kuvassa on kattoon lisätty roofGable operaattori. Operaattori vaatii kolme parametria joilla määrätään katon korkeus sekä molempien reunojen pituus. Tämän jälkeen katto täytyi jakaa kahteen pintaan; päällimmäiseen ja sivuihin. Päällimmäiselle pinnalle määrättiin oma tekstuuri ja sivuille seinää vastaava tekstuuri. Katon tekstuuri on kopioitu samasta harjoituksesta kuin ikkunaobjekti.

```
katto-->
roofGable(20, 0.2, 0.2)
comp(f) (top: kattoTekstuuri | side : seinä)

kattoTekstuuri -->
setupProjection(0, scope.xy, scope.sx, scope.sy)
projectUV(0)
texture("buildings/roofs/flatroof4.bw.jpg")
```

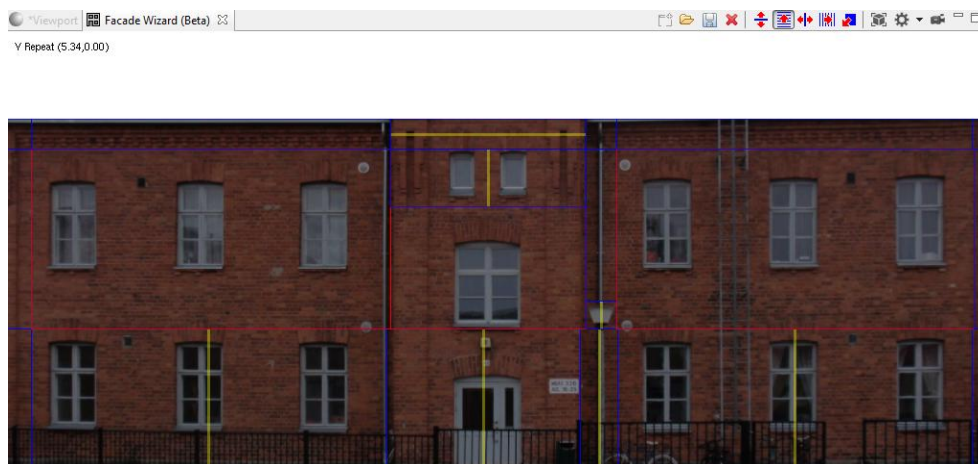


**KUVA 19. Katon lisääminen malliin**

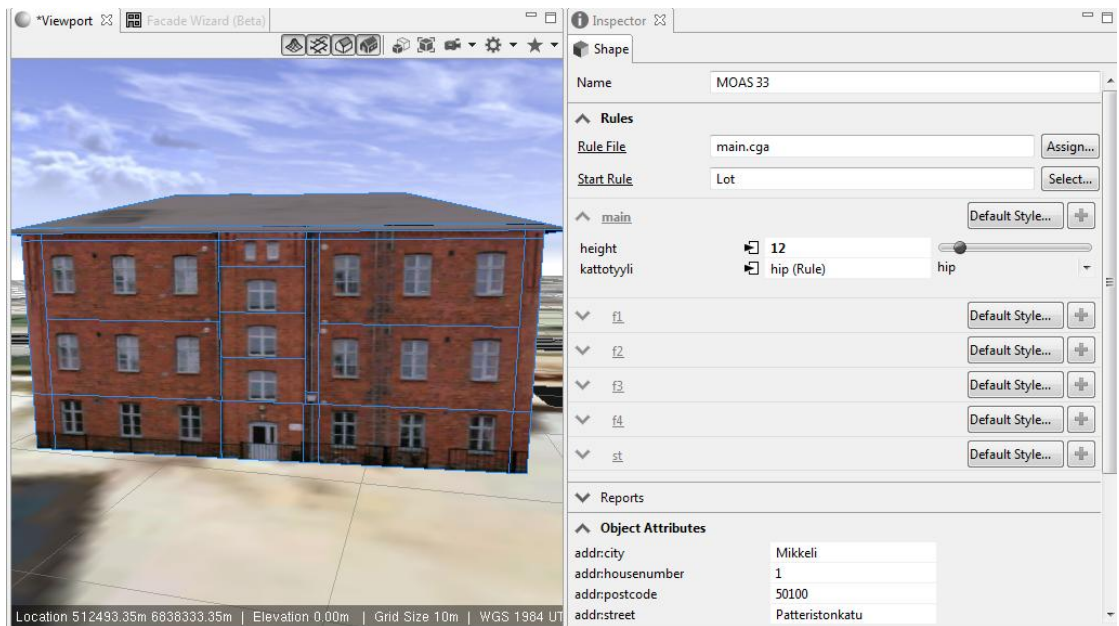
### 3.1.2 Kuvan perusteella mallintaminen

Ohjelmalla on myös mahdollista luoda sääntöjä kuvien perusteella Facade wizardin avulla. Tällä toiminnolla saadaan luotua nopeasti CGA sääntötiedosto yhdestä rakennuksen seinästä. Facade Wizardin avulla voidaan määritellä esimerkiksi mitkä kerrokset pitää toistaa jos rakennuksen korkeus nousee sekä mitkä pinnat pysyvät vakiona. Tällaisilla vakiona pysyvillä pinnoilla tarkoitan esimerkiksi ovien kohtaa.

Näiden sääntöjen luominen tapahtuu viivoja asettelemalla. Sinisillä viivoilla määritellään mitkä kerrokset pysyvät muuttumattomina ja keltaisilla viivoilla kerrotaan, että pinta ei saa venyä. Punaisilla viivoilla, joita kutsutaan toistuviksi viivoiksi, kerrotaan ohjelman toistamaan kyseiset pinnat jos rakennuksen korkeus tai leveys muuttuu. Näiden viivojen havainnollistaminen on esitetty kuvassa 20. Tässä kuvassa ainoastaan ylimmäisen kerroksen on tarkoitus toistua jos rakennuksen korkeus muuttuu. Kuvassa 21 nähdään, kuinka rakennuksen ylin kerros toistunut, kun rakennuksen korkeus muuttuu.



**KUVA 20. Julkisivukuvan työstäminen Facade Wizardilla**



**KUVA 21. Säännön mukautuminen rakennuksen korkeuteen nähden**

Kuvassa 22 on esitetty kuinka näitä sääntöjä voidaan käyttää rakennuksen mallintamiseen import operaattorilla. Jotta jokaiselle pinnalle saatiin määrättyä eri sääntö, täytyi rakennuksen seinät jakaa ensimmäiseksi omiin pintoihin comp(f)-operaattorilla. Tämän jälkeen seinälle voitiin määrätä sääntö kutsumalla attribuuttia (f1) ja viittaamalla Facade alkuperäisiin tiedoston Facade-juureen.

```
import f1 : "gen_Facade_etuseina.cga"
import f2 : "gen_Facade_takaseina.cga"
import f3 : "gen_Facade_jun_seina.cga"
import f4 : "gen_Facade_mik_seina.cga"

Lot-->
  extrude(korkeus) rakennus

rakennus-->
  comp (f) {front : etuseina |
           back  : takaseina |
           left  : vasen  |
           right : oikea  |
           top   : katto}

etuseina-->
  f1.Facade

takaseina-->
  f2.Facade

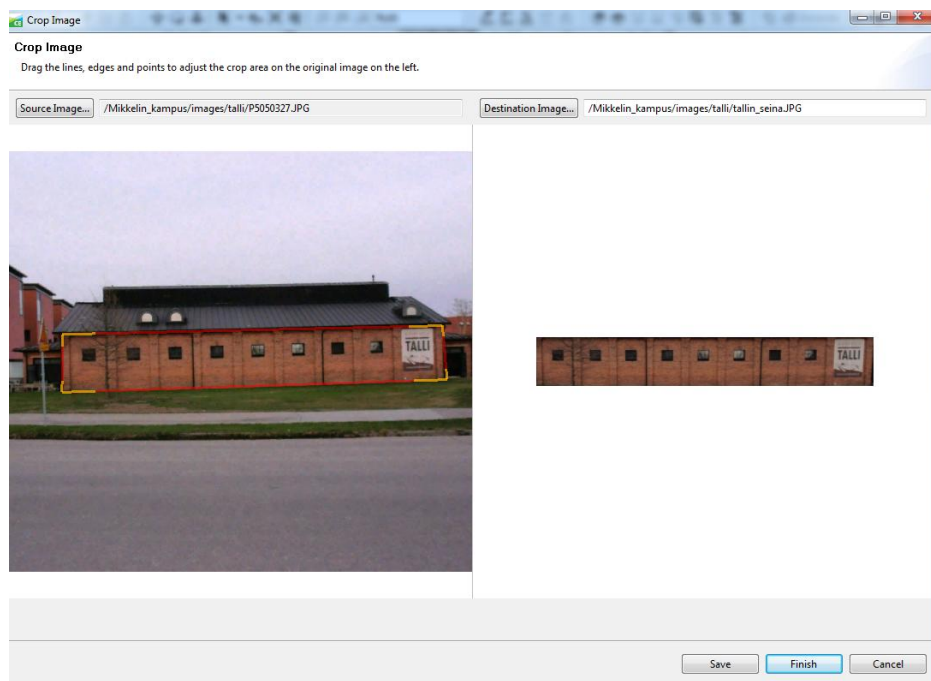
vasen-->
```



**KUVA 22. Sääntöjen tuominen sääntöön import komennolla**

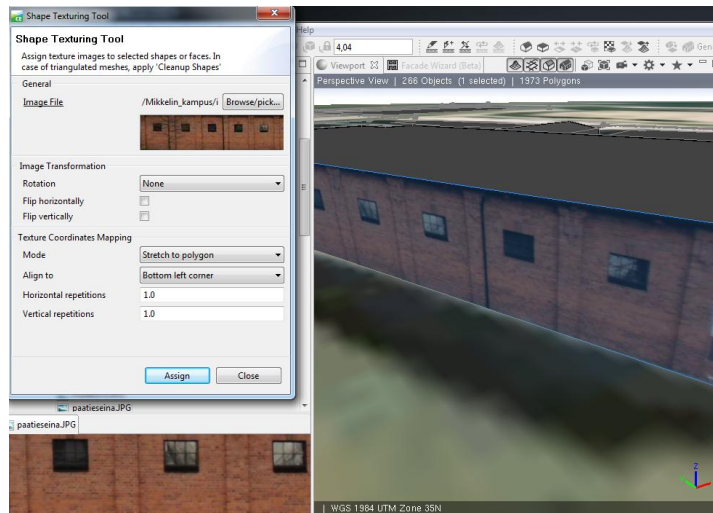
### 3.1.3 Mallintaminen piirtämällä

Crop image on CityEnginen kuvankäsittely työkalu. Tällä työkalulla luodaan tekstuurimaisia kuvia aidoista valokuvista. Ongelmia, joita tämän työn käyttämisestä saattaa esiintyä, olivat ohjelman kaatuminen. Varsinkin nopeasti työskenteleminen tällä työkalulla aiheutti usein ohjelman kaatumisen. Ilmeisesti työkalu vaatii koneelta paljon ja siksi suosittelenkin käyttämään tätä työkalua rauhallisesti, ellei sitten omista todella tehokasta tietokonetta. Seuraavassa kuvassa 23 on esitetty työkalu sekä vasemmalle oleva kuva on rajattu niin että oikealle jää vain rakennuksen seinä.



**KUVA 23. Tekstuurin tekeminen Crop image työkalulla**

Kuvien käsittelyn jälkeen nostin ravintola Tallin seinät pystyy polygonal shape creation -työkalun avulla. Helpoin tapa minun mielestä tällaisen rakennuksen teksturointiin on siirtää kuva tai tekstuuri suoraan sille tarkoitetulle pinnalle. Kuvassa 24 on esitetty kuvien asettaminen sekä shape texture -työkalu, joka avautuu kun kuva liitetään pinnalle. Tämän työkalun voi myös avata erikseen. Työkalulla voidaan esimerkiksi tuoda tai vaihtaa tekstuuri sekä muuttaa kuvan asettelua niin, että kuva asettuu pinnalle tietyissä mitoissa. Kuvassa 24 tekstuuri on asetettu niin, että se venyy pinnan mukaisesti.



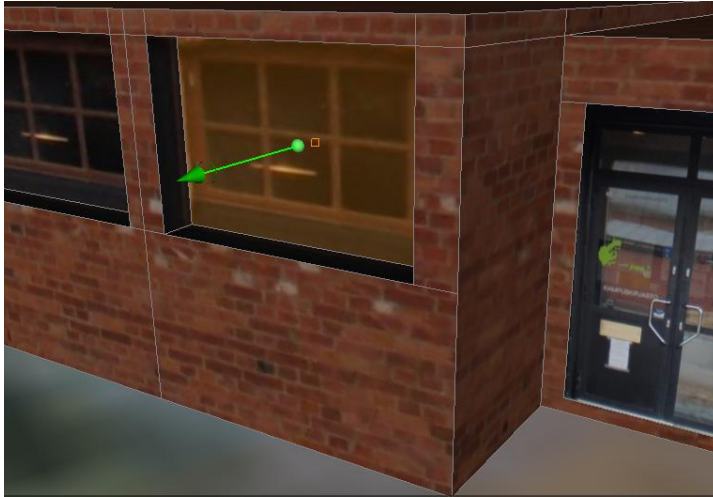
**KUVA 24. Tekstuurin asettaminen malliin shape texturing -työkalulla**

Tekstuuriin jälkeen nostin vielä harjakaton esiin sekä lisäsin vielä rectangular shape creation -työkalulla katon lisäosan. Lisäosan tein erikseen ja liitin tämän rakennukseen yhdistämällä mallit toisiinsa. Kuvassa 25 on mallinnettu ravintola Talli valmiiksi. Ongelmia, joita tämän rakennuksen mallintamisessa esiintyivät, olivat ainoastaan ohjelman kaatuminen ja tämäkin johtui todennäköisesti heikosta koneesta.



**KUVA 25. Ravintola Talli valmiina**

Kokeilin myös kuinka polygonal shape creation -työkalu toimii kokonaisen rakennuksen mallintamiseen. Kirjaston mallintaminen oli hieman samanlaista kuin Tallin. Tämän mallin teksturoimiseksi en kuitenkaan käyttänyt kokonaisia seinäkuvia vaan loin crop image-työkalulla alkuperäisistä kuvista erikseen seinä-, ikkuna- ja ovitekstuureja. Lisäsin myös tähän malliin enemmän muotoja kuin ravintola Talliin. Esimerkiksi kuvassa 26 on esitetty kuinka kirjaston ikkunoihin on lisätty syvyyttä polygonal-työkalulla.



**KUVA 26. Ikkunan pinnan syventäminen polygonal shape creation työkalulla**

Työkalulla yksinkertaiset rakennukset onnistuvat melko kätevästi, mutta mitään kovinkaan tarkkoja malleja en lähtisi näin tekemään. Malliin voidaan kylläkin liittää erilaisia objekteja sääntöjen avulla joita voidaan tehdä muun muassa tehdä ohjelmalla tai esimerkiksi SketchUpilla. Näiden objektien tuominen malliin jäi tässä työssä selvittämättä. Ongelmia esiintyi muun muassa välillä pintojen lisäämisen kanssa. Näiden lisääminen saattoi joskus aiheuttaa toisen pinnan kadottamisen. Tämän sai kuitenkin korjattua joko palaamalla taaksepäin tai tekemällä uuden seinän. Kaikkien vastoin käymisten jälkeen sain kuitenkin mallinnettua ja teksturoitua kirjaston, kuten kuvasta 27 nähdään.



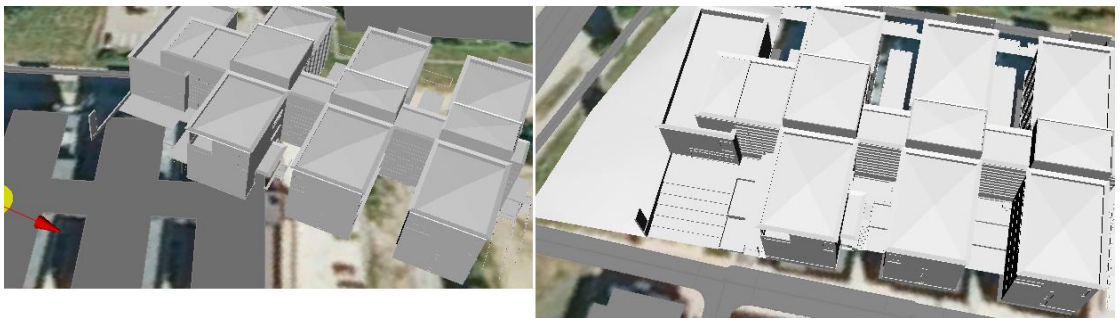
**KUVA 27. Kirjasto mallinnettu valmiiksi**

Kirjaston mallintamisen perusteella, suosittelen monimutkaisten rakennusten mallintamiseen esimerkiksi SketchUpia. Tässä työssä kokeilin kuinka SketchUpilla mallinnettu Mikpoli voidaan tuoda CityEnginen näkymään. Mikpoli on Mikkelin ammattikorkeakoulun opiskelijoiden mallintaminen.

Helpoin tapa mallin tuomiseen on muuttaa malli SketchUpissa colladaan muotoon ja valita tiedoston tallennuskohteeksi CityEnginen projekti. Tämä siitä syystä, että SketchUp tallentaa mallin lisäksi myös tekstuurit samaan kansioon. Näin saadaan estettyä ongelmat mallin teksturoinnin suhteen.

Collada muotoiset tiedostot voidaan myös tuoda CityEnginee import-toiminnon avulla tai kopioimalla tiedostot suoraan projektikansioon. Tässä tavassa täytyy kuitenkin muistaa myös tuoda mallissa käytetyt tekstuurit projektikansioon. Ongelmia, joita tässä menetelmässä saattaa esiintyä, ovat tekstuurien puuttuminen. Tämä johtuu siitä, että collada-tiedostossa tekstuurien polku on väärä. Collada-päätteiset tiedostot ovat kuitenkin tekstimuotoisia ja näin ollen tekstuurin polun voi esimerkiksi korjata CityEnginen tekstieditorissa.

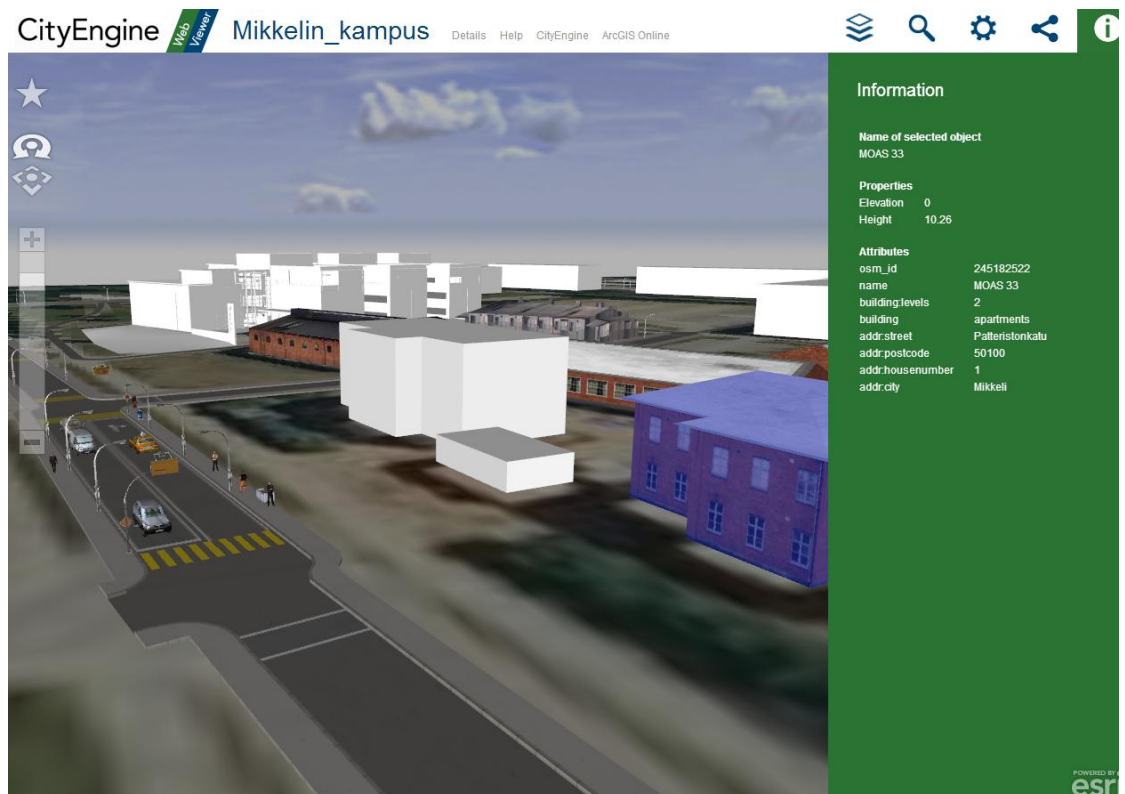
CityEngineen tuotujen mallien tiputtaminen näkymään onnistuu esimerkiksi vetämällä mallin omalle paikalleen. Collada-tiedostoja voidaan käyttää myös säännöissä esimerkiksi samoin kuin ikkunaobjekteja. Tällaisen mallin tekemisessä on kuitenkin hyvä muistaa asettaa malli origoon, jotta vältetään epämieluisilta sijoitteluongelmilta. Sitä en kerinnyt miettimään, voitaisiinko kokonaisia rakennuksia käyttää säännöissä rakennuksien muodostamiseen. Säännöissä voitaisiin esimerkiksi määrittellä, että tietyt SketchUpilla mallinnetut rakennukset asettuisivat attribuuttien avulla omille paikoilleen. Mikpolin tiputtamisen jälkeen, mallin joutui vielä siirtämään ohjelman siirtotyökaluilla omalle paikalleen. Kuvassa 28 vasemmalla Mikpoli on tiputettu näkymään ja oikealla malli on asetettu omalle paikalleen.



**KUVA 28. Vasemmalle Mikpoli on tuotu näkymään ja oikealla Mikpoli on asetettu oikealle paikalle**

### 3.2 Web viewer

Ohjelman web viewer -toiminnolla malleja voidaan katsella selaimella. Ennen kuin mallit voidaan viedä selaimelle, täytyy nämä muuttaa CityEnginen web scene-muotoon. Tätä varten mallit täytyy valita esimerkiksi selection-työkalulla, jonka jälkeen nämä muunnetaan ohjelman export models -toiminnolla cityengine web scene -muotoon. Web scene -muotoon muuttaessa valitaan muun muassa tiedoston sijainnin paikka sekä mitkä kaikki tasot halutaan näyttää selaimella. Tämän jälkeen ohjelma luo määrättyyn kansioon 3WS tiedoston, joka voidaan avata offline-tilassa web vieweriin. 3WS tiedosto voidaan myös julkaista verkossa, jolloin CityEnginää ei tarvita mallien katsomiseen selaimella. Julkaiseminen kuitenkin vaatii ArcGIS-käyttäjätunnukset. Ongelmia joita web scenessä ilmeni, olivat kirjaston tekstuurien vääristyminen. Tälle ongelmalle en valitettavasti löytänyt vastausta. Kuvassa 29 mallit näytetään selaimelle offline-tilassa. Vihreällä pohjalla näkyy myös rakennuksen tiedot, jotka ovat peräisin OSM-tiedostosta.



**KUVA 29. Web Viewer offline-tilassa**

## 4 YHTEENVETO

Tämän opinnäytetyön tavoitteena oli tutkia, kuinka CityEngine soveltuu Mikkelin ammattikorkeakoulun rakennusten mallintamiseen sekä minkälaisia erilaisia menetelmiä ohjelma tarjoaa niiden luomiseen. Työssä perehdyttiin myös kuinka OpenStreetMap-tietoa voidaan käyttää ohjelmassa ja lopuksi vielä selvitettiin, kuinka mallit voidaan näyttää selaimella käyttäen ohjelman Web viewer -toimintoa.

Työn edetessä kohtasin useita erilaisia ongelmia. Nämä ongelmat johtuivat usein siitä, että olin aloittelija ohjelman suhteen. Aluksi ongelmiin löytyi vastauksia harjoitusten sekä ohjelman help sivuston kautta, mutta työn mentäessä pidemmälle jouduin turvautumaan CityEngine-foorumiin.

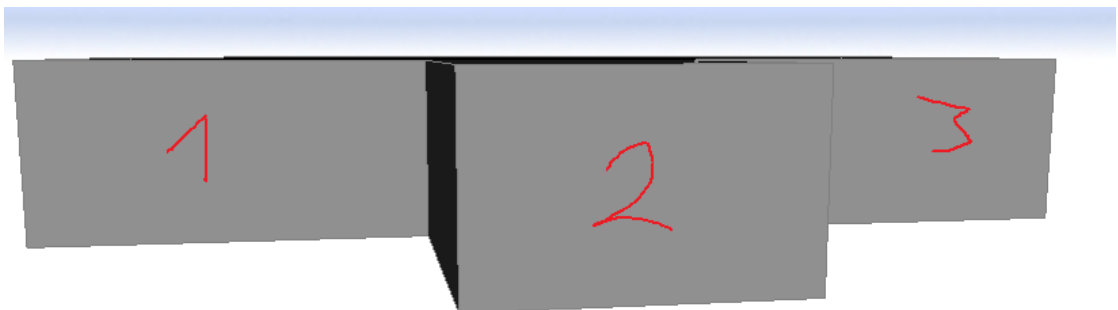
Ongelmia esiintyi myös joidenkin ohjelman toimintojen kanssa. Esimerkiksi web viewer vääristää jostain kumman syystä kirjaston tekstuurit. Yritin selvittää, mistä tämä johtuu, mutta en löytänyt vastausta tähän. Toisinaan joillekin pinnoille ei voinut asettaa shape texture -työkalulla tekstuuria ennen kuin näkymän oli avannut uudelleen. Välillä tämäkään ei auttanut ja ohjelman joutui käynnistämään uudelleen. Ohjelma myös vaatii koneelta paljon, sillä ohjelma kaatui useita kertoja työtä tehdessäni.

Omasta mielestä OpenStreetMap-tiedosto toimi hyvin CityEnginessä alueen katuverkkona. Katuverkosto oli kattava sekä OSM-tiedosto sisälsi myös paljon elementtejä kampuksista kuten esimerkiksi kaikki kampuksen rakennukset. Tästä haluaisin kiittää alueen tekijöitä. Harmillista kuitenkin oli, että rakennuksille ei löytynyt korkeuksia vaan jouduin itse lisäämään nämä ohjelmassa.

Rakennusten mallintamiseen paras menetelmä oli omasta mielestä ehdottomasti sääntöihin perustuva menetelmä. Tämä kuitenkin vaatii aikaa sekä facade wizardia käyttäessä myös paljon kuvia. Tekstuurien liittäminen tiputtamalla toimi hyvin yksittäisiin rakennuksiin, mutta useampiin rakennuksiin tämä on omasta mielestä liian työläs menetelmä. Monipuolisten ja erittäin tarkkojen mallien luomiseen käyttäisin mieluummin esimerkiksi SketchUpia kuin ohjelman polygonal shape creation -työkalua. SketchUpilla on kuitenkin omasta mielestä mukava ja helppo mallintaa sekä mallien tuominen CityEngineen tekstuureineen onnistuu vaivattomasti. Polygonal shape creation työkalulla oli omasta mielestä hankala luoda esimerkiksi ikkunoiden paikkoja, sillä jouduin

piirtämään kaikki viivat erikseen. Myös ylimääräisten viivojen poistaminen tuotti välillä ongelmia. Mallin liittäminen selaimelle onnistui myös omasta mielestä helposti. Kummallista kuitenkin oli, että kirjaston tekstuuri vääristyi selaimelle vaikka CityEnginen näkymässä tekstuurit oli normaalisti.

Omasta mielestä yksi varteenotettava jatkotutkimus olisi perehtyä syvemmin sääntömenetelmän saloihin. Itselleni esiintyi suureksi ongelmaksi selvittää kuinka kuvan(30) mukaiselle rakennukselle voidaan luoda sellainen sääntö, että pinnalle yksi, kaksi ja kolme saataisiin erilaiset muodot. Samaan tutkimuksessa voitaisiin myös perehtyä ohjelmassa käytettyyn python ohjelmointikieleen, sillä tässä työssä kyseinen kieli jäi kokeilematta. Python ohjelmointikielellä voidaan esimerkiksi luoda scriptejä, joilla on mahdollista animoida rakennuksen mallintaminen tai luoda kansio, johon on tallennettu esimerkiksi näkymässä käytetty katuverkosto sekä talot. Tällainen tiedoston luomiscripti on kätevä kun halutaan tuoda yhteen kansioon kaikki tarvittavat tiedostot. Ohjelmasta voisi myös tehdä toisenlaisen tutkimuksen, jossa perehdytään kuinka Arcmapissa käytetty aineistoa voidaan käyttää CityEnginessä katuverkostojen ja rakennusten mallintamiseen.



### **KUVA 30. Rakennuksen muodon havainnollistaminen**

Ohjelman tulevaisuus näyttää minun mielestä valoisalta. Tästä kertoo esimerkiksi se, että CityEngine foorumeille uusimpien keskustelujen aloituspäivämäärä ei ole enää parin vuoden takaisin. Omasta mielestä uusia keskusteluja aloitetaan jatkuvasti kun taas vuoden alussa jouduin lukemaan parin vuoden takaisia keskusteluja. CityEngine on myös nykyään integroitu ArcGIS alustaan helpottamaan ohjelmien kommunikointia keskenään. Integroitumisen myötä kaupungit voivat ladata CityEnginen käyttöönsä Esrin kuntalisenssillä. Tämän lisenssin myötä uskon, että CityEngine tulee menestymään myös Suomessa. Tässä voi kuitenkin mennä vielä vuosia, sillä ohjelma on hyvin

nuori verrattuna muihin kaupunkimallinnus ohjelmiin. Mikkelin ammattikorkeakoulu on kuitenkin jo aloittanut ohjelman edistämisen siten, että tämä tuo ohjelman ainakin opiskelijoiden tietoisuuteen. Koulu on nimittäin asentanut ohjelman joillekin koneille ja aikovat jatkossa lisätä tämän myös opetukseen. Kaiken kaikkiaan ohjelma oli hauska työkalu ja odotan innolla mitä ohjelma tulee edistymään jatkossa.

## LÄHTEET

3D-visualisoinnille paljon käyttökohteita. 2011. Esri Finland asiakaslehti 1, 33.

Aho, Anne 2009. Google Mapsin käyttö karttapalvelun toteutuksessa. Laurea-ammattikorkeakoulu. Tietojenkäsittelyn koulutusohjelma. Opinnäytetyö. 8.2009.

ArcGIS 10 monipuolistaa ja tehostaa 3D-mallinnusta. 2011. Esri Finland asiakaslehti 1, 33.

ArcGIS on kokonaisjärjestelmä. 2011. Esri Finland asiakaslehti 2, 15.

Autodesk Revit 2014. Autodesk Revit rakennussuunnitteluun ja rakentamiseen. WWW-dokumentti. <http://www.autodesk.fi/products/autodesk-revit-family/features/software-for-bim/all/gallery-view>. Päivitetty 2014. Luettu 23.1.2014.

BlindSquare 2014. About BlindSquare. WWW-dokumentti. <http://fi.blindsquare.com/about/>. Päivitetty 2014. Luettu 3.4.2014.

BlomURBEX™ integroitu lukuisiin paikkatieto-ohjelmistoihin. 2012. Blom. WWW-dokumentti. [http://newsletter.blomasa.com/newsletter/2012/june/finland/june\\_fi\\_4.htm](http://newsletter.blomasa.com/newsletter/2012/june/finland/june_fi_4.htm). Päivitetty 4.6.2012. Luettu 15.3.2014.

Erving, Anna 2007. Julkisivutekstuurin liittäminen 3D-malliin. Helsingin Teknillinen Korkeakoulu. Tekniikan ylioppilas. Diplomityö.

Esri CityEngine 2014. Esri CityEngine. PDF-dokumentti. [http://www.esri.fi/midcom-serveattachmentguid-1e29d1baa9b8cea9d1b11e2aafe7376cb18a082a082/cityengine\\_bro\\_suomeksi\\_lores.pdf](http://www.esri.fi/midcom-serveattachmentguid-1e29d1baa9b8cea9d1b11e2aafe7376cb18a082a082/cityengine_bro_suomeksi_lores.pdf). Ei päivitystietoja. Luettu 13.1.2014.

Esri Finland 2012. Mitä ovat paikkatieto ja GIS. WWW-dokumentti. [http://www.esri.fi/referenssit/mita\\_paikkatieto\\_on](http://www.esri.fi/referenssit/mita_paikkatieto_on). Päivitetty 4.12.2012. Luettu 27.3.2014.

Esri tarjoaa mullistavan ohjelmiston 3D-suunnitteluun. 2011. Esri Finland asiakaslehti 2, 20–21.

Google 2014a. Kolmiulotteisten kuvien katseleminen Earth-näkymässä. WWW-dokumentti. [https://support.google.com/maps/answer/3092441?hl=fi&ref\\_topic=3092425](https://support.google.com/maps/answer/3092441?hl=fi&ref_topic=3092425). Päivitetty 2014. Luettu 23.3.2014.

Google 2014b. Earth-näkymä. WWW-dokumentti. <https://support.google.com/maps/answer/3093427>. Päivitetty 2014. Luettu 23.3.2014.

Google 2014c. What is Google Map Maker. WWW-dokumentti. [https://support.google.com/mapmaker/answer/157176?hl=fi&ref\\_topic=1093469&rd=](https://support.google.com/mapmaker/answer/157176?hl=fi&ref_topic=1093469&rd=). Päivitetty 2014. Luettu 27.3.2014.

- Google Map Maker. 2014. Wikipedia. WWW-dokumentti.  
[http://en.wikipedia.org/wiki/Google\\_Map\\_Maker](http://en.wikipedia.org/wiki/Google_Map_Maker). Päivitetty 4.4.2014. Luettu 5.4.2014.
- Gröger, Gerhard, Kolbe, Thomas H., Czerwinski, Angela & Nagel, Claus. 2008. OpenGIS® City Geography Language (CityGML) Encoding Standard. OpenGIS® Standard.
- Haggrén, Henrik 2002. Ortokuvien tuottaminen. WWW-dokumentti.  
<http://foto.hut.fi/opetus/220/luennot/7/7.html>. Päivitetty 11.10.2002. Luettu 27.2.2014.
- Isotalo, Katri 2013. Kaupunkimalli on muutakin kuin visualisointia. *Positio* 1, 17–19.
- Karttojen perusominaisuudet. 2014. Paikkaoppi. WWW-dokumentti.  
[http://www.paikkaoppi.fi/Oppitunnit\\_ja\\_projektimallit/Oppituntikonaisuudet/1.1](http://www.paikkaoppi.fi/Oppitunnit_ja_projektimallit/Oppituntikonaisuudet/1.1). Ei päivytystietoja. Luettu 25.2.2014.
- Kaupunkimalli on tulevaisuuden kantakartta. 2012. Vianova. WWW-dokumentti.  
<http://www.vianova.fi/Uutiset/Finland/Kaupunkimalli-on-tulevaisuuden-kantakartta#.UsoFc9JdUrU>. Päivitetty 7.3.2012 Luettu 27.2.2014.
- Keskikiikonen, Hannes 2012. Jokamiesluokan kartoitusta. *Positio* 3, 14–16.
- Kouvola 2013. Kantakartta. WWW-dokumentti.  
<http://www.kouvola.fi/index/aikuisvaestolle/asuminenjaymparisto/kartatjapaikkatietopalvelut/karttatuotteet/kantakartta.html>. Päivitetty 21.11.2013. Luettu. 3.3.2014
- Kuhno, Mikko 2012. Kaupunkialueen 3D-mallintamisen kevyet menetelmät. Mikkelin Ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö. 30.11.2012
- Laitinen, Tuomas 2012. Paikkatietoaineistojen jakelun uudistaminen Helsingin kaupungin kaupunkimittausosastolla. Metropolia Ammattikorkeakoulu. Maanmittaustekniikan koulutusohjelma. Insinööri. 4.4.2012.
- Lehtovirta, Pekka & Nuutinen, Kari 2000. 3D-sisältötuotannon peruskirja. Jyväskylä. Dosendo Finland.
- Lentokuva 2014. Ilmakuvaus ja palvelut. WWW-dokumentti.  
<http://www.lentokuva.fi/index.php/ilmakuvauspalvelut/palveluista>. Päivitetty 2014. Luettu 16.3.2014.
- Luebke, David, Reddy, Martin, Cohen, Jonathan D., Varshney, Amitabh, Watson, Benjamin & Huebner, Robert 2002. *Level of Detail for 3D Graphics*. Elsevier Science & Technology.
- Löytönen, Markku, Toivonen, Tuuli. & Kankaanrinta, Ilta-Kanerva 2003. *Globus GIS paikkatietojärjestelmä*. Porvoo. WSOY.
- Maanmittauslaitos 2014. Avoimien aineistojen tiedostopalvelu. WWW-dokumentti.  
<http://www.maanmittauslaitos.fi/aineistot-palvelut/latauspalvelut/avoimien-aineistojen-tiedostopalvelu/>. Ei päivytystietoja. Luettu 28.3.2014.

- Mikkeli 2013. 3D-malli Mikkelin keskustasta. WWW-dokumentti.  
<http://www.mikkeli.fi/palvelut/3d-malli-mikkelin-keskustasta>. Päivitetty 6.11.2013.  
Luettu 15.3.2014.
- Micro Aided Design 2014. ArchiCAD. WWW-dokumentti.  
<http://www.mad.fi/mad/archicad.html>. Ei päivitystietoja. Luettu 23.1.2014.
- Nieminen, Susanne 2011. Länsi-Uudenmaan vesi ja ympäristö ry:n Hiidenvesi-hankkeen paikkatietojärjestelmän suunnittelu. Novia Yrkeshögskolan. Luonnonvara-ala. Opinnäytetyö. 12.5.2011.
- Novapoint Virtual Map 2014. Vianova. PDF-dokumentti.  
[http://www.vianova.fi/content/download/279/45463/version/115/file/VirtualMap\\_FIN\\_WEB%5B1%5D.pdf](http://www.vianova.fi/content/download/279/45463/version/115/file/VirtualMap_FIN_WEB%5B1%5D.pdf). Ei päivitystietoja. Luettu 27.2.2014.
- OpenStreetMap. 2014. Wikipedia. WWW-dokumentti.  
<http://en.wikipedia.org/wiki/OpenStreetMap>. Päivitetty 14.2.2014. Luettu 20.2.2014.
- Paikkatietoa on kaikkialla. 2014. Paikkaoppi. WWW-dokumentti.  
[http://www.paikkaoppi.fi/Oppitunnit\\_ja\\_projektimallit/Oppituntikonaisuudet/2.1](http://www.paikkaoppi.fi/Oppitunnit_ja_projektimallit/Oppituntikonaisuudet/2.1). Ei päivitystietoja. Luettu 25.2.2014.
- Rajala, Janne 2012. Virtuaalimalli ja -tila suunnitteluvälineenä. Aalto-yliopisto. Pääsuunnittelija -koulutusohjelma. Ammatillisen kehittymisen raportti. 11.2012
- Sani, Ilari 2009. OpenStreetMap on karttojen Wikipedia. MikroPC. WWW-dokumentti.  
<http://www.mpc.fi/neuvot/openstreetmap+on+karttojen+wikipedia/a333480>. Päivitetty 3.10.2009. Luettu 4.2.2014.
- Saarentaus, Miranda 2000. Satelliittikuvat paikkatiedoissa – Miksi ja miten. Maankäyttö 5. Verkkolehti.  
[http://www.maankaytto.fi/arkisto/mk500/mk500\\_373\\_saarentaus.pdf](http://www.maankaytto.fi/arkisto/mk500/mk500_373_saarentaus.pdf). Päivitetty 18.12.2000. Luettu 12. 3.2014.
- Suomisto, Jarmo 2013. Kaupunkien tietomallit Euroopassa. PDF-dokumentti.  
[http://www.forumvirium.fi/sites/default/files/fiksu\\_kalasadama\\_12\\_12\\_2013.pdf](http://www.forumvirium.fi/sites/default/files/fiksu_kalasadama_12_12_2013.pdf). Päivitetty 20.11.2013 Luettu 28.3.2014.
- Tuhola, Esa & Kristiina, Viitanen 2008. 3D-mallintaminen suunnittelun apuvälineenä. Jyväskylän. Tammertekniikka.
- Tukholma kehittyi kolmiulotteisena. 2013. Blom. WWW-dokumentti.  
[http://newsletter.blomasa.com/newsletter/2013/September/fi/September\\_fi\\_5.htm](http://newsletter.blomasa.com/newsletter/2013/September/fi/September_fi_5.htm). Päivitetty 9.5.2013. Luettu 1.4.2014.
- Ultirender 2014. Vectorworks Architect. WWW-dokumentti.  
[http://www.ultirender.com/pages/ohjelmistot/nemetschek\\_na/vectorworks\\_architect.htm](http://www.ultirender.com/pages/ohjelmistot/nemetschek_na/vectorworks_architect.htm). Päivitetty 2014. Luettu. 17.4.2014.

Vesseli 2014. Paikkatieto on keino tutkia, havainnollistaa ja hallita ympäristöä ja yhteiskuntaa käsittelevää alueellista tietoa. WWW-dokumentti.

<http://www.vesseli.fi/paikkaoppi/abc.htm>. Ei päivitystietoja. Luettu 25.2.2014.

Visualisoi menestys Autodeskin infra-alan tietomallinnusratkaisulla. 2013. Vianova. PDF-dokumentti.

[http://www.vianova.fi/content/download/3128/56270/version/95/file/infrastructure\\_design\\_suite\\_2013.pdf](http://www.vianova.fi/content/download/3128/56270/version/95/file/infrastructure_design_suite_2013.pdf). Päivitetty 2013. Luettu 27.2.2014.