Niko Siltakorpi

Sisällöntuotannon menetelmiä 3D-peliympäristöön

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

13.4.2014

Metropolia

Tekijä Otsikko Sivumäärä Aika	Niko Siltakorpi Sisällöntuotannon menetelmiä 3D-peliympäristöön 38 sivua + 1 liite 13.4.2014
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Mediatekniikka
Suuntautumisvaihtoehto	Digitaalinen media
Ohjaaja	Lehtori Antti Laiho

Insinöörityön tarkoituksena oli kehittää sisällöntuotannon menetelmiä ja prosesseja 3D-peliympäristöön. Menetelmien ja prosessien kehittämisessä keskityttiin ensisijaisesti 3D-peliympäristössä toimivien 3D-mallien ja 360°-panoraamakuvien tuotantomenetelmiin.

Työn tavoitteena oli tutkia erilaisia tuotantomenetelmiä, jotka voisivat tehostaa ja nopeuttaa 3D-mallien tuotantoa. 360°-panoraamakuvien tuotannossa tärkein tavoite oli kehittää toimiva tuotantoprosessi, jolla 360°-panoraamakuvia pystyisi tuottamaan käytettäväksi 3D-peliympäristössä.

3D-mallien tuotannon tehostamiseen ja nopeuttamiseen etsittiin erilaisia ratkaisuja ohjelmista, jotka pystyvät tuottamaan 3D-malleja valokuvien avulla ja 3D-skannaamalla. Pyrkimyksenä oli löytää edullinen ratkaisu, jonka avulla pystyttäisiin tuottamaan laadukas 3D-malli mahdollisimman nopeasti. 360°-panoraamakuvien tuotantoprosessin kehitykseen kuului oikean valokuvaustekniikan löytäminen, 360°-panoraamakuvan tuottaminen valokuvista ja toimivan käyttötavan löytäminen 3D-peliympäristössä.

Tutkimuksista ja kokeiluista huolimatta 3D-peliympäristöön tarkoitettujen 3D-mallien tuotantomenetelmään ei löydetty täysin tavoiteltua ratkaisua. 3D-mallinnusohjelmistoilla työskentely todettiin tehokkaimmaksi ja laadukkaimmaksi tavaksi tuottaa 3D-malleja 3D-peliympäristöön. 360°-panoraamakuvien tuotantoon kehitetty tuotantoprosessi täytti tavoitteet ja vaatimukset, kun oikeat tuotantotavat, laitteet ja ohjelmat löytyivät.

3D-mallien tuotannon tehostamiseen ja nopeuttamiseen voitaisiin löytää ratkaisuja ammattikäyttöön tarkoitetuista ohjelmistoista ja kehittyneimmistä laitteista, jotka pystyisivät vähentämään käyttäjän roolia 3D-mallinnuksessa. 360°-panoraamakuvien tuottamiseen on kehitetty useita toimivia ja helppokäyttöisiä ratkaisuja, mutta tulevaisuudessa kehitetään varmasti vielä edullisempia, helpompia ja laadukkaampia ratkaisuja.

Avainsanat	3D, pelimoottori, 3D-skannaus, panoraamakuva

Author	Niko Siltakorpi
Title	Content production methods for 3D game environment
Number of Pages	38 pages + 1 appendix
Date	13 April 2014
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructor	Antti Laiho, Lecturer

The subject of this thesis was to develop content production methods and processes for a 3D game environment. Procedures and processes focused on the development of a 3D game environment, working with 3D models and 360° panoramic production methods.

The aim was to examine a variety of production methods that could enhance and accelerate the production of 3D models. In 360° panoramic images, the production of the main objective was to develop a manufacturing process by which 360° panoramic images could be used to produce a 3D game environment.

To speed up the production efficiency of 3D models, a variety of solutions programs which can produce 3D models of photographs and 3D scanning were studied. The requirement was to find a low-cost solution for high-quality 3D models for proper and fast production.

In spite of studies and experiments, no proper solution for 3D models in the gaming environment was found. 3D modeling software working was found to be the most effective and high quality way to produce 3D models in a 3D game environment.

The results of this study showed that suitable solutions to make the production of 3D models more efficient and faster can be found. These solutions will also reduce the role of the user in 3D modelling. Hopefully even more efficient ways for the production will be developed in future.

Keywords	3D, game engine, 3D scanning, panorama

Sisällys

Lyhenteet ja käsitteet

API	Application Programming Interface. Sovellusohjelmien kieli- ja viestintätapa kommunikoida käyttöjärjestelmän tai muun valvontaohjelman kanssa.
CAD	Computer Aided Design. Tietokoneavusteinen suunnittelu.
FBX	Autodesk-yrityksen kehittämä tiedonsiirtojärjestelmä, jonka avulla pystytään siirtämään informaatiota 3D-ohjelmistoissa ja -työkaluissa.
Pelimoottori	Ohjelma, joka vastaa objektien mallintamisesta ja piirtämisestä näytölle.
Polygoni	Monikulmio
SDK	Software development kit. Joukko ohjelmistojen kehitystyökaluja, jotka mahdollistavat sovelluksen kehityksen erilaisille ohjelmistoille, laitteille ja käyttöjärjestelmille.

1 Johdanto

Insinöörityön taustalla on työelämälähtöinen ongelma, johon pyritään löytämään ratkaisuja kokeilemalla ja kehittämällä entistä parempia tuotantomenetelmiä. Parempien tuotantomenetelmien avulla puolestaan halutaan tehostaa sisällöntuotannon prosesseja ja työskentelymenetelmiä. Sisällöntuotantoa tehostamalla työtehtävien toivotaan valmistuvan nopeammin, mikä vapauttaisi aikaa tuotannosta muihin tehtäviin ja parantaisi ylipäätään tuotannon kannattavuutta yrityksessä. Sisällöntuotannon paranneltavat työvaiheet ovat 3D-mallien ja 360°-panoraamakuvien tuotanto, sillä näiden työvaiheiden on todettu kuormittavan eniten sisällöntuotantoa.

Sisällöntuotannossa huomataan eniten resursseja kuluvan huonekalujen ja rakennuksien 3D-mallinnuksen työtehtäviin. Näiden työtehtävien toteuttamiseen halutaan löytää tehokkaampia tuotantomenetelmiä. Erilaisten ohjelmien ja tekniikoiden avulla 3D-mallinnuksen työtehtävistä olisi mahdollista saada automatisoidumpia ja tehokkaampia. Lisäksi arvioidaan, että tulevaisuudessa tuotantoon saapuvat asiakasprojektit voivat ruuhkauttaa tuotantoa, eivätkä nykyiset resurssit enää todennäköisesti riitä toteuttamaan projekteja aikataulussa.

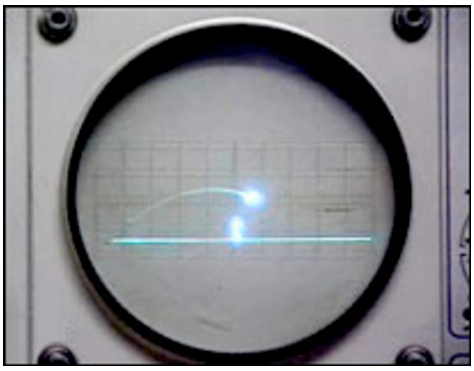
360°-panoraamakuvien tuottamiseen puolestaan on tavoitteena kehittää tuotantoprosessi, joka olisi mahdollisimman lineaarinen ja automatisoitu. Tuotantoprosessiin kehitetään ilmakuvausmenetelmä: tapa yhdistää valokuvat yhdeksi 360°-panoraamakuvaksi. Tämä taas on toimiva käyttötapa 3D-peliympäristössä. 360°-panoraamakuvan tuotantoprosessin kehityksessä on kuitenkin haasteita, sillä tehtävä on uudentyyppinen, ja lopputuloksen odotetaan olevan laadukas.

2 3D-peliteknologia

Pelimoottorit, 3D-grafiikka ja 3D-piirron rajapinnat ovat tärkeitä osa-alueita 3D-peliteknologiassa. Pelimoottoreita avuksi käyttäen pelinkehittäjät pystyvät keskittymään paremmin pelin kehitykseen ja sen toteuttamiseen. Tehokkaat pelimoottorit tarjoavat työkaluja, joiden avulla 3D-peliteknologiaa pystytään käyttämään hyödyksi nopeammin ja pienemmällä vaivalla. Pelimoottorit tuovat 3D-grafiikkaa käyttäen peleihin ja virtualisointeihin aivan uusia ulottuvuuksia verrattuna 2D-grafiikkaan. 3D-grafiikan piirtämisestä huolehtivat erilaiset rajapinnat, jotka toimivat erilaisissa laitteissa ja käyttöjärjestelmissä.

2.1 Pelimoottorit

Ensimmäisen videopelin kehittivät Thomas T. Goldsmith Jr. ja Estle Ray Mann vuonna 1947. Pelin nimi oli Cathode-Ray Tube Amusement Device (ks. kuva 1). Pelissä pelaaja yrittää ampua säteellä yläpuolella liikkuvaan pisteeseen. Jos piste saa osuman, siitä seuraa simuloitu räjähdys. Tästä lähtien pelien ja pelimoottoreiden kehitys on edennyt nopeasti kohti fotorealistista grafiikkaa, tarkkoja fysiikoita ja elämyksellisiä pelikokemuksia. (Milian & Chan 2012.)



Kuva 1. Cathode-Ray Tube Amusement Device -pelin monitori (Brown 2010).

Useat, jotka ovat olleet tekemisissä pelialan kanssa tai ovat seuranneet pelialaa, ovat voineet kuulla puhuttavan pelimoottoreista. Kaikki eivät kuitenkaan välttämättä heti ensi kuulemalta tiedä, mitä pelimoottorilla tarkoitetaan. Pelimoottoria voi verrata vaikkapa autonmoottoriin: se vie peliä eteenpäin ja pitää sen käynnissä. Jos pelimoottorissa on vikoja, peli voi jossakin vaiheessa rikkoutua – aivan kuten autokin. Yksinkertaistettuna

pelimoottori ajaa pelin perustoimintoja, esimerkiksi peliin liittyviä tehtäviä, grafiikan renderöintiä ja fysiikkaa ja vastaanottaa komentoja. Pelimoottorin avulla pelin kehittäjät ja artistit pystyvät keskittymään itse pelin tuottamiseen ja kehittämiseen.

Pelimoottorit tarjoavat komponentit ja ominaisuudet, joiden avulla peli herää henkiin. Latausikkuna, elementtien esilletuonti, animaatiot, objektien törmäily, fysiikat, komentojen vastaanotto, graafinen käyttöliittymä ja jopa tekoäly ovat yleensä pelimoottorin peruskomponentteja. Kuitenkin pelin sisältö, 3D-mallit, tekstuurit, tarina ja se, miten asiat käyttäytyvät pelissä, luovat itse pelin. Komponentteja voi verrata jälleen vaikka auton ominaisuuksiin: auton muotoilu, korin väri, vanteet, pakoputki ja istuimet ovat osa auton sisältöä ja tekevät autosta ainutlaatuisen.

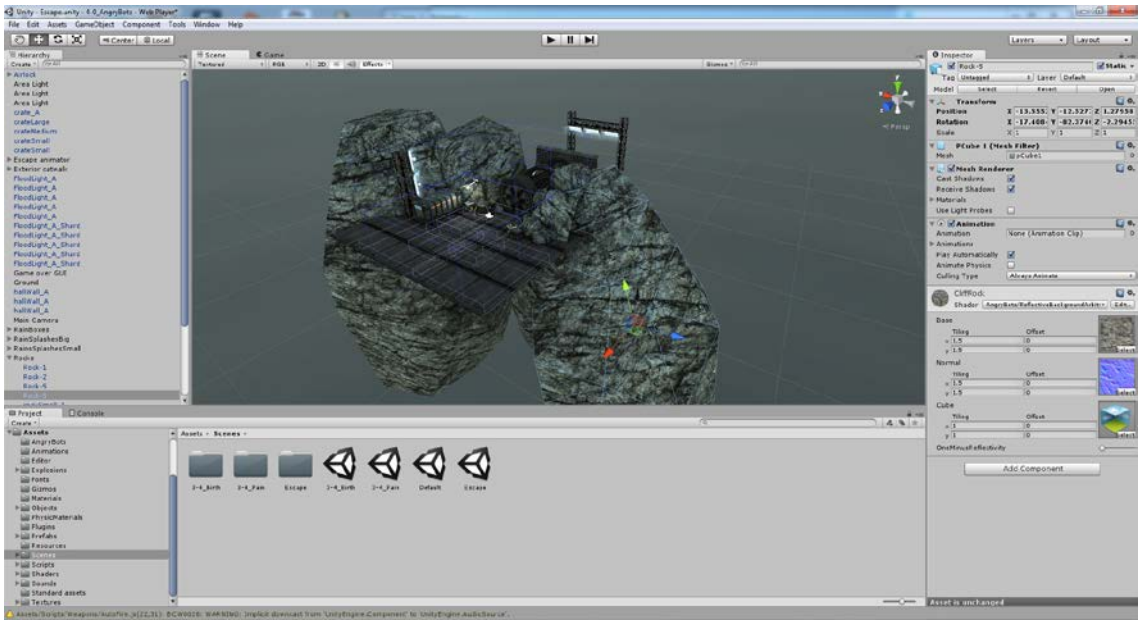
API ja SDK ovat kaksi termiä pelialalla, jotka ovat yhteydessä pelimoottoriin. API on sovelluksen rajapinta, jonka avulla käyttöjärjestelmää, kirjastoja ja palveluita pystytään käyttämään hyödyksi. SDK on kokoelma kirjastoja, API puolestaan rajapintoja ja työkaluja, jotka tekevät ylipäätään mahdolliseksi ohjelmoida käyttöjärjestelmille ja palveluille. Useat pelimoottorit tarjoavat API-rajapinnat SDK:nsa mukana. Esimerkiksi Unreal Engine -pelimoottori tarjoaa ohjelmointirajapinnan, jonka avulla peliä voidaan kehittää UnrealScript-skriptikielen ja kirjastojen kautta. Ohjelmointirajapinta ja muut työkalut saadaan käyttöön hankittaessa pelimoottorin lisenssi.

Monet pelitalot tekivät pitkään oman pelimoottorinsa ja pitivät sen yrityksen sisällä. Ongelmaksi nousi tällöin pelimoottorin jatkuva kehittäminen, koska tietokoneet ja laitteet parantuivat jatkuvasti. LucasArtsin SCUMM- ja Sierran SCI-pelimoottorit olivat suosittuja etenkin seikkailupelien pelimoottoreina aina 1980-luvun lopusta 1990-luvun puoleenväliin. Vuosien myötä pelimoottoreiden kehittämisen kustannukset ovat kuitenkin nousseet merkittävästi. Useat pelialan yritykset ovat nykyään erikoistuneet pelkästään pelimoottorin tai komponenttien kehittämiseen – ennemmin kuin kokonaisen pelin tekemiseen. (Ward 2008.)

Pelimoottoreita on markkinoilla paljon erilaisia ja erityyppisiä, ja niillä voi tuottaa erityyppisiä pelejä erilaisille alustoille. Pelimoottoreilta vaaditaan paljon, koska tehokkaat laitteet mahdollistavat kauniin ja yksityiskohtaisen grafiikan, mutta mobiililaitteet taas rajoittavat kokonaisuutta. Monipuolisimmilla pelimoottoreilla pystyy tekemään pelejä monille alustoille ja laitteille. Vaikka pelimoottoreita on suuri valikoima, on markkinoilla kuitenkin muutamia johtavia pelimoottoreita. (Enger 2013.)

Internetisssä on verkkosivustoja, joissa on listattu tämän hetken suosituimpia pelimoottoreita. Pelimoottorit on listattu käyttäjien ja trendin mukaan. Suosituimpien pelimoottorien kärjessä ovat Unity3D, CryENGINE 3, Source ja Unreal Engine 3. (100 Most Popular Engines Today 2014.)

Insinöörityössä käytettiin Unity3D-pelimottoria (ks. kuva 2), mikä on otettu huomioon 3D-sisällöntuotannon menetelmissä. Unity3D on monipuolinen ja helposti lähestyttävä pelimoottori, jonka suosio on kasvanut viime vuosien aikana. Unity3D tukee monia laitteita ja alustoja, mikä tekee siitä kilpailukykyisen pelimoottorin moneen eri käyttötarkoitukseen. Unity3D-ilmaisversiolla on mahdollista tehdä ja julkaista pelejä, mutta proversiossa on vielä paremmat ominaisuudet. Kirjoitushetkellä Unity3D-pelimoottorin proversio maksaa 1 140 € tai 57 € /kk. (Unity3D store 2014.)



Kuva 2. Unity3D-pelimoottorin editori.

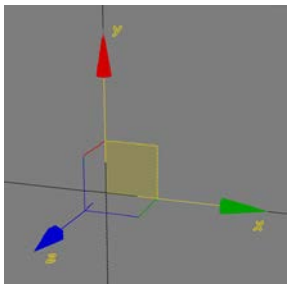
2.2 3D-grafiikka

3D-grafiikasta on tullut korvaamaton apuväline monenlaisissa suunnittelu- ja visualisointitehtävissä. Kolmiulotteinen malli on yksi monipuolisimmista tavoista esittää tai visualisoida tavaroita, rakennuksia ja erilaisia asioita. Monissa projekteissa käytetään hyväksi 3D-malleja asioiden visualisoimiseen tai simulaatioon, ennen kuin projektia viedään pidemmälle. Rakennusalalla 3D-visualisoinnit tuntuvat olevan jo ehdoton kri-

teeri, ja erilaiset 3D-simulaatiot ovat hyödyllisiä tutkimus- ja opetuskäytössä. 3D soveltuu myös viihdealalle, koska sen avulla voidaan toteuttaa asioita, jotka olisivat muuten mahdottomia tai kalliita toteuttaa. 3D-grafiikkaa käytetäänkin peleissä, elokuvissa ja mainonnassa nykyään paljon. Oppimiseen ja hyötykäyttöön tarkoitetut pelit ja sovellukset ovat selvästi nouseva liiketoiminnan ala. (Puhakka 2008: 24.)

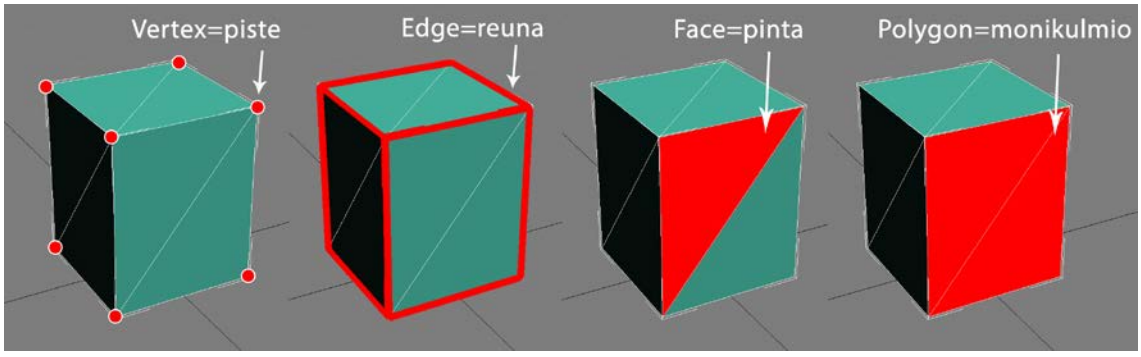
3D-malli

3D-mallit ovat vektorigrafiikkaa, ja ne koostuvat pisteistä, joilla on kolme ulottuvuutta. Pisteitä yhdistävät viivat ja pinnat, jotka luovat muodot 3D-mallille. 3D-grafiikan luontiin tarvitaan XYZ-koordinaatisto (ks. kuva 3), joka kertoo pisteiden sijainnit.



Kuva 3. XYZ-koordinaatisto.

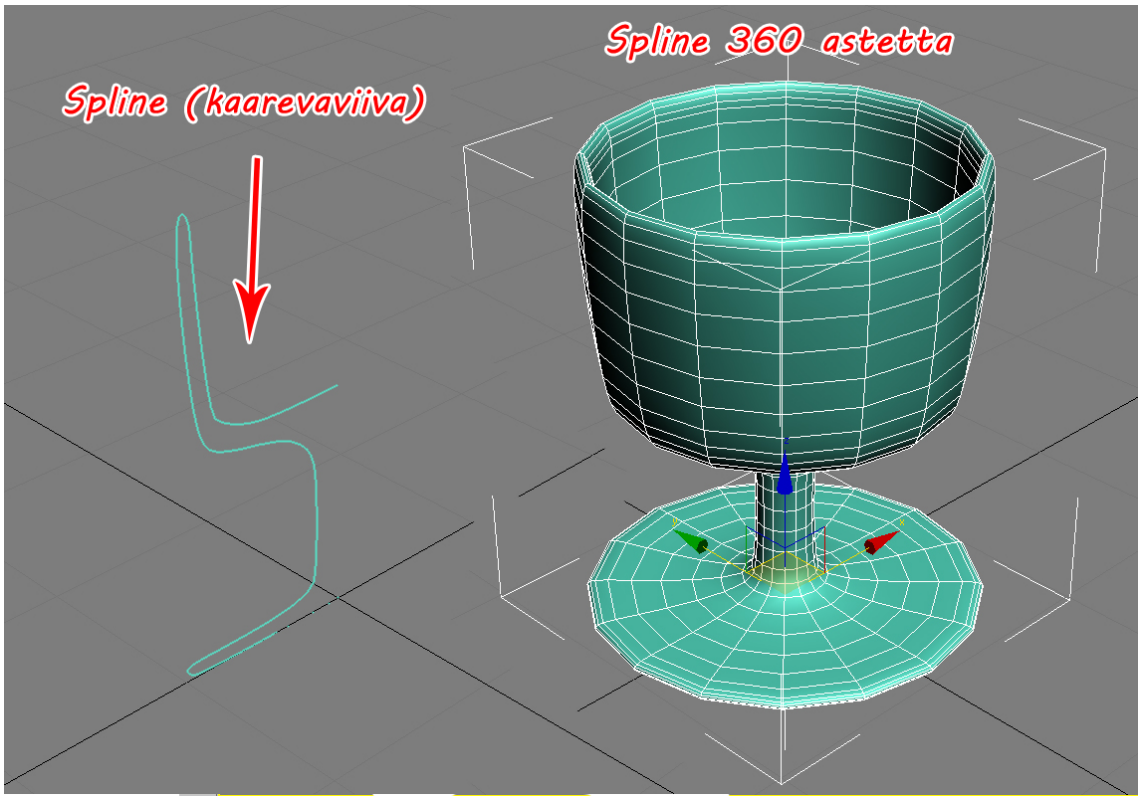
X-akseli kulkee vasemmalta oikealle, Y-akseli alhaalta ylös, ja Z-akseli antaa syvyyden. Itse malli koostuu muutamasta elementistä. Vertex on yksi piste, jolla on X-, Y- ja Z-arvot. Vertexien väleissä olevia viivoja kutsutaan nimellä *”edge”*. Kun kolmen vertexin välissä on pinta, sitä kutsutaan termillä *”face”*. Yksi polygoni rakentuu kahdesta pinnasta, jotka muodostavat neliön (ks. kuva 4). (Puhakka 2008: 30–31.)



Kuva 4. 3D-mallin rakennuselementit.

3D-Spline-mallinnus

Spline-mallinnus tunnetaan myös nimellä nurbs-mallinnus, ja se on tehokas vaihtoehto 3D-mallintamiseen. 3D-Spline-mallinnus kuvaa kaarevaa viivaa. Kaarevalle viivalle on määritelty useita muokkauskohtia. Kaarevalla spline-viivalla voidaan sorvaamalla tai puristamalla luoda 3D-geometriaa (ks. kuva 5). (Harper 2012: 118–119.)



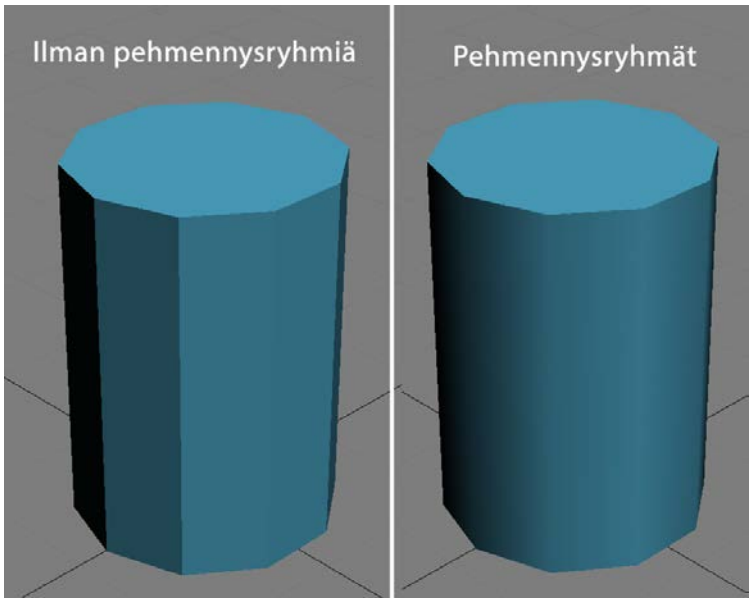
Kuva 5. Esimerkki Spline-mallinnuksesta.

Mesh-mallinnus

Yleisin 3D-mallin työstötapa on polygonimallinnus. Kun luodaan Mesh-mallia, se varastoi eri elementtejä. Näitä ovat vertexit (pisteet), edget (reunat), face (pinta) ja polygonit (monikulmiot). Useissa sovelluksissa tallennetaan vain vertexit (pisteet), edget (reunat) ja joko facet (kasvot) tai polygonit (monikulmiot). Pisteiden sijainnin lisäksi mallilla on muitakin tietoja, kuten väri, normaali vektori ja tekstuurin koordinaatit.

Pinnoista koostuvalle Mesh-mallille asetetaan yleensä pehmennysryhmiä (ks. kuva 6), jotka heijastavat valon pehmeästi mallin pinnalta. Vaihtoehtoisesti mallille voi asettaa geometrisiä pehmennysryhmiä, jotka taas lisäävät pintoja. Erittäin korkealla pintojen

määrällä on vähemmän alueita, jotka vaativat pehmennysryhmiä. Niiden pinnat ovat niin pieniä, että tarve on merkityksetön. Yksi vaihtoehto on vielä yksinkertaisesti irrottaa pinta-alueita Mesh-mallista. Erillisissä Mesh-malleissa ei voida asettaa yhtenäisiä pehmennysryhmiä.

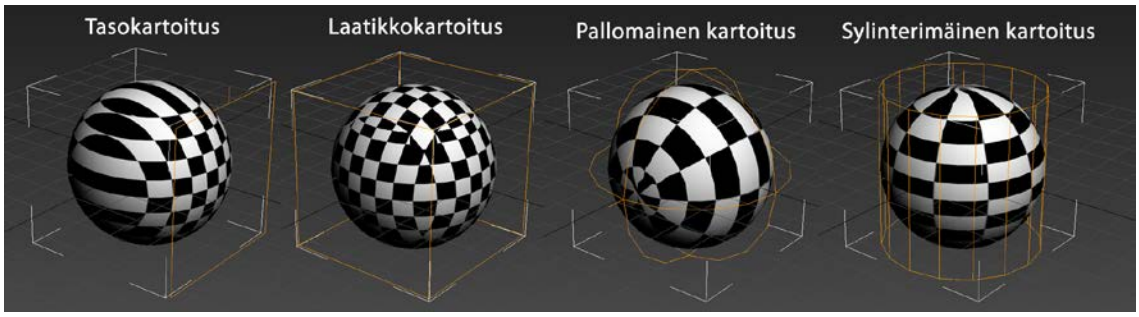


Kuva 6. Mesh-mallin pehmennysryhmien erot.

Mesh-muodossa voi määritellä myös muita hyödyllisiä tietoja. Joihinkin ryhmiin voidaan määritellä erillisiä elementtejä, joissa on osaobjekteja, luuston animaatioita tai erillisiä toimijoita. Materiaalit määritellään, jotta eri osat käyttävät erilaisia tekstuureja ja pinta-ominaisuuksia. Kaikilla pinnoilla on UV-koordinaatit, jotka kertovat, miten materiaali asettuu. (Harper 2012: 214.)

Teksturointi

3D-mallille on mahdollista asettaa tekstuuri. UV-kartoitus (mapping) on helpoin tapa asettaa tekstuuri 3D-mallin pinnalle. Tekstuurin lisäksi voidaan puhua nahasta (skin). Erona on se, että nahka on luotu jotain tiettyä mallia varten ja se on monimutkaisempi, esimerkiksi olio tai ampuma-ase. Tekstuuria voidaan pitää yksinkertaisempana materiaalina, kuten ruoho, lattia tai seinä. UV-kartoituksen tekeminen tasaisille pinnoille on helpompaa kuin monimuotoiselle oliolle. (Ahearn 2006: 60.)



Kuva 7. Erilaisia UV-kartoitusmenetelmiä.

Seuraavaksi tarkastellaan muutamia erilaisia UV-kartoitusmenetelmiä. Tasokartoitus (planar mapping) toimii kuin projektori. Kartoitus projisoidaan suoraan tietystä suunnasta, joten se sopii parhaiten tasaisille pinnoille, kuten seiniin ja lattioihin. Laatikkokartoitus (box mapping) asettaa kartoituksen suoraan kuudesta suunnasta. Se sopiikin nimensä mukaisesti parhaiten laatikkomaisiin malleihin. Pallomainen kartoitus (spherical mapping) ympäröi pyöreästi mallin joka puolelta; se sopiikin juuri pallomaisiin ja elliptisiin muotoihin. Sylinterimäinen kartoitus (cylindrical mapping) ympäröi mallin, mutta päistä kartoitus on suoraan kuten tasossa – se sopiikin sylinterin muotoisiin malleihin (ks. kuva 7). (Ahearn 2006: 61–64.)

2.3 3D-piirron rajapinnat

Yleisimmät 3D-piirtämiseen tarkoitetut rajapinnat ovat nimeltään DirectX ja OpenGL. DirectX on Microsoftin kehittämä Windows-käyttöjärjestelmiin sidoksissa oleva rajapinta. OpenGL on avoin ja alustasta riippumaton. (Puhakka 2008: 355.)

DirectX

Vuonna 1994 Microsoft kehitti uutta käyttöjärjestelmää, Windows 95:tä. Microsoftin peliohjelmoinnin alustana oli tuolloin DOS. Se nähtiin hyvänä alustana, koska se pääsee vaikuttamaan suoraan näytönohjaimiin, hiireen, näppäimistöön ja äänilaitteisiin. Windows 95:ssä oli rajoitettu pääsy komponentteihin, koska se käytti muistin suojausta. Microsoft halusi tarjota ohjelmoijille samanlaisen tavan päästä vaikuttamaan laitteisiin uuden käyttöjärjestelmän Windows 95:n avulla. (The History of DirectX 2010.)

Syyskuussa 1995 ensimmäinen versio DirectX:stä julkaistiin. Vuoden 1995 jälkeen julkaistiin uusi versio DirectX:stä vähintään kerran vuodessa. Vuonna 2002 Microsoft julkaisi DirectX 9:n, ja tämän julkaisun mukana oli Shader-versio 2.0. Microsoft on jatkanut DirectX 9:n säännöllistä päivitystä. Vuonna 2004 yritys julkaisi DirectX 9.0c:n, joka tukee Shader Model 3.0:aa. DirectX 10 -versio oli saatavilla uusiin Windows Vista- ja 7-käyttöjärjestelmiin. DirectX 11 on rajapinnan uusin versio, ja se pystyy käyttämään tehokkaasti moniytimisiä suorittimia ja grafiikkaprosessoreja, mikä parantaa piirto- ja prosessointinopeutta videopeleissä. (St. John 2013.)

Open GL

Vuonna 1980 ohjelmiston kehittäminen oli haaste, etenkin jos halusi ohjelman tukevan monenlaisia näytönohjaimia. Ohjelmistojen kehittäjien täytyikin kirjoittaa omat ajurinsa kuhunkin laitteistoon. Heidän täytyi myös käsitellä paljon erilaisia rajapintoja, mikä teki tehtävästä entistä vaikeamman. Toinen ongelma oli se, että jokainen kehitystiimi joutui kirjoittamaan ajurit samoihin laitteistoihin. Tuloksena oli paljon samaa lähdekoodia, jota kirjoittivat useat kehittäjät.

Vuoden 1990 alussa Silicon Graphics oli johtava yritys, joka mahdollisti 3D-grafiikan työasemille. Yritys käytti työasemillaan ohjelmointirajapintaa nimeltä IRIS GL. Liikesalaisuutenaan IRIS GL käytti SGI-tekniikkaa eikä avointa standardia. Ohjelmointirajapinta katsottiin helpoksi käyttää, ja sillä oli hyviä ominaisuuksia. Tuolloin kilpailevat yritykset, kuten Sun Microsystems, IBM ja Hewlett-Packard, olivat myös tuomassa 3D-laitteistoja markkinoille. Ne puolestaan käyttivät ohjelmointirajapintaa nimeltä PHIGS.

Kun muut toimittajat toivat toisen ohjelmointirajapinnan 3D-markkinoille, Silicon Graphicsin markkinaosuus pieneni. Vaikuttaakseen markkinoihin Silicon Graphics päätti tehdä IRIS GL:stä version avoimeen lähdekoodistandardiin. Silicon Graphics loikin uuden rajapinnan nimeltä OpenGL, joka perustuu IRIS GL:ään. Vuonna 1992 Silicon Graphics oli johtava OpenGL-arkkitehtuurin kehittäjä. Nykyään yhdeksällä yrityksellä on äänioikeus OpenGL:n kehittämiseen. Kaikkien erilaisten ohjelmistojen toimintoja ei oteta suoraan käyttöön OpenGL-versioihin.

Onneksi näytönohjainvalmistajat voivat tarjota OpenGL-laajennuksia. Näillä laajennuksilla on mahdollista käyttää eri ohjemistojen ominaisuuksia. Jos ominaisuuksia käyttä-

vät useat valmistajat, laajennukset voivat tulla viralliseksi lisäksi OpenGL-standardia. OpenGL:n tuleviin versioihin tuleekin yhä enemmän ja enemmän vaikutteita pelin kehittäjiltä ja pelialalta. (The History of OpenGL 2010.)

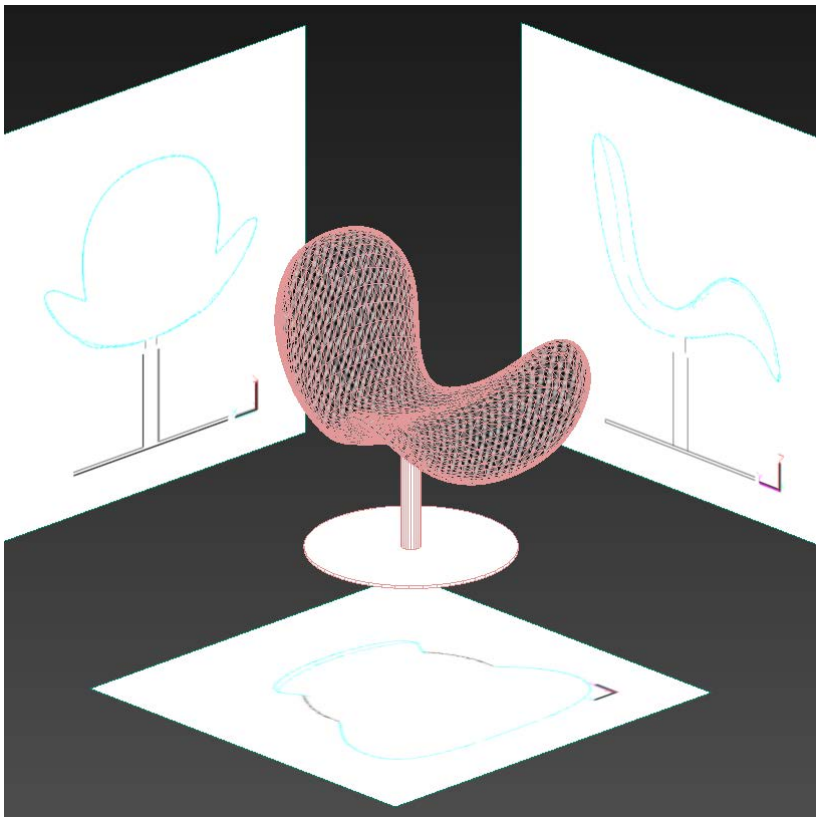
3 3D-mallien tuotantomenetelmät

On olemassa erilaisia 3D-mallinnusmenetelmiä ja ohjelmistoja, joilla voidaan tuottaa 3D-malleja erilaisilla tekniikoilla, laitteilla ja lähdemateriaaleilla. 3D-mallien laatu ja rakenne voi olla erilainen riippuen käyttäjän tai ohjelman luomasta geometriasta. 3D-mallinnusmenetelmät ja ohjelmistot vaikuttavat myös 3D-mallin tuottamisen tehokkuuteen ja nopeuteen. Insinöörityössä tutkittiin ohjelmia, joilla voi tuottaa 3D-malleja valokuvien avulla ja 3D-skannaamalla. Nämä tuotantomenetelmät eroavat tietyissä määrin 3D-mallinnusohjelmistojen tuotantomenetelmästä, jossa 3D-malli tuotetaan spline- ja mesh-mallinnustapoja käyttäen manuaalisesti.

3.1 3D-mallinnusmenetelmiä 3D-peliympäristöön

3D-mallinnusmenetelmät valokuvien avulla ja 3D-skannaamalla voivat tuoda 3D-mallintamiseen tehokkuutta ja helppoutta ohjelmien tuomalla automatiikalla. Tuotosten jälkeen on kuitenkin huomattu, että automatiikkaa käyttävillä ohjelmilla luotua 3D-mallia voi joutua korjaamaan ja optimoimaan 3D-mallinnusohjelmistolla 3D-peliympäristöön sopivaksi. Tavoitteena olikin tutkia erilaisten 3D-mallinnusmenetelmien tehokkuutta, helppoutta ja laatua. 3D-mallinnusmenetelmien tärkeimpinä tuotantokohteina olivat huonekalut, ihmiset ja rakennukset. Kaikki edellä mainitut objektit ovat muotonsa ja rakenteensa takia hieman erilaisia 3D-mallinnettavia.

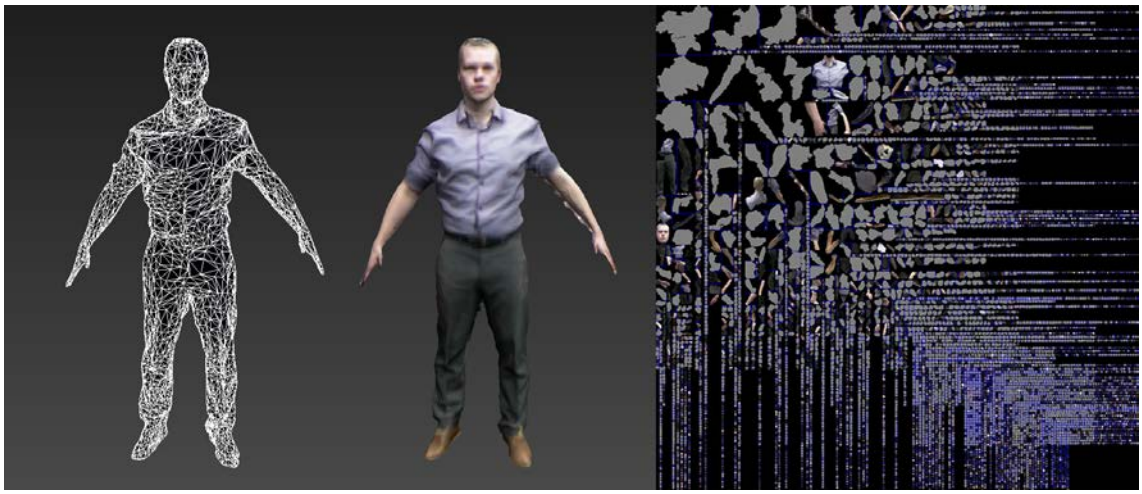
Huonekalut voivat olla monimuotoisia tai yksinkertaisia, isoja tai pieniä. Useissa tapauksissa huonekalujen 3D-mallinnukseen voi ottaa avuksi valokuvia ja viivapiirroksia (ks. kuva 8). Vaikka huonekaluja voi olla helppo ja nopea mallintaa 3D-mallinnusohjelmistolla, se ei ole välttämättä tehokkain ratkaisu – etenkin, jos huonekaluja täytyisi mallintaa satoja tai jopa tuhansia. Tässä tapauksessa tehokas ja hyvä tuotantoprosessi tai automatisointi voi nopeuttaa työtä huomattavasti. Nykyään huonekaluvalmistajilla on mahdollisesti valmiita 3D-malleja huonekaluista, ja internetissä on olemassa laaja valikoima jopa ilmaisia 3D-malleja (Cad-symbolit 2013; Tan 2013).



Kuva 8. 3D-malli huonekalusta, jonka ympärillä näkyvät objektin muodot ja mitat viivapiirroksena.

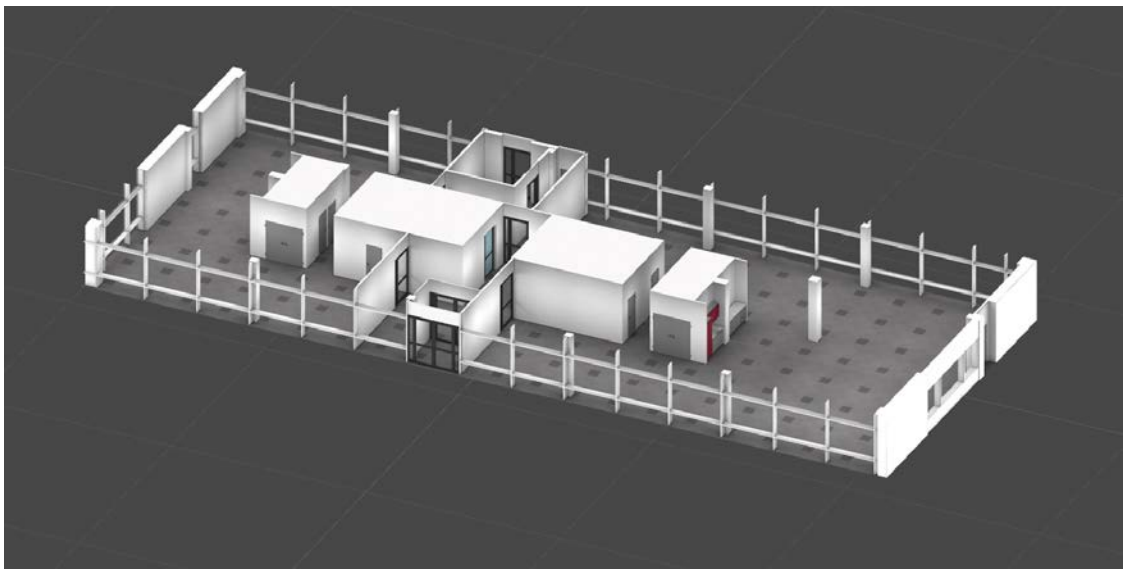
Orgaaniset 3D-mallit, esimerkiksi ihmiset ja eläimet, voivat olla haastavia mallintaa, mikäli mallin on tarkoitus olla mahdollisimman realistinen ja tunnistettava. Orgaaniset objektit voivat olla haastavia mallintaa, koska niissä voi olla paljon yksityiskohtia ja erilaisia pinnan muotoja. Etenkin ihmisen mallintaminen realistiseksi on hankalaa, koska ihmissilmä huomaa helposti pienetkin epäkohdat ja virheet. Työtä ja taitoa vaativat myös orgaanisen 3D-mallin teksturointi ja UV-kartoitus, puhumattakaan animaatioista. Orgaaniset objektit koostuvat usein erilaisista pinnan värityksista ja materiaaleista, joiden tekeminen hyvin voi olla haastavaa. (Realistic Head modeling guide 2007.)

Hyvällä automaatiotekniikalla on kuitenkin mahdollista saada kohtuullisen hyvä ihmismalli valmiiksi teksturoituna, mutta ongelmia voi esiintyä erityisesti optimoinnissa. 3D-skannatun 3D-mallin rakenne ja tekstuuri eivät ole aina ihanteellisia 3D-peliympäristöön, sillä ne sisältävät liikaa informaatiota ja sisältö on liian monimutkaista (ks. kuva 9).



Kuva 9. 3D-skannattu 3D-malli ihmisestä ja prosessissa syntynyt tekstuuri eivät ole ihanteellisia 3D-peliympäristöön. 3D-mallin geometria on sekava ja tekstuuri koostuu liian monista osista.

Rakennuksen 3D-mallintaminen voi olla yksinkertaista, mikäli rakennus koostuu tasaisista pinnoista ja suorista kulmista (ks. kuva 10). Rakennukset voivat olla kuitenkin suuria, ja niissä voi olla paljon yksityiskohtia. Rakennuksen 3D-mallintamista helpottavat tarkat pohjapiirrokset, valokuvat ja pintamateriaalien ohjeistukset. Rakennusten suunnitteluun ja 3D-mallintamiseen on kehitetty CAD-sovelluksia, joiden avulla pystytään käsittelemään rakennuksia kaksi- ja kolmiulotteisesti sekä tekemään kokonaisvaltainen suunnitelma rakennuksesta (Lehtovirta & Nuutinen 2000: 118).

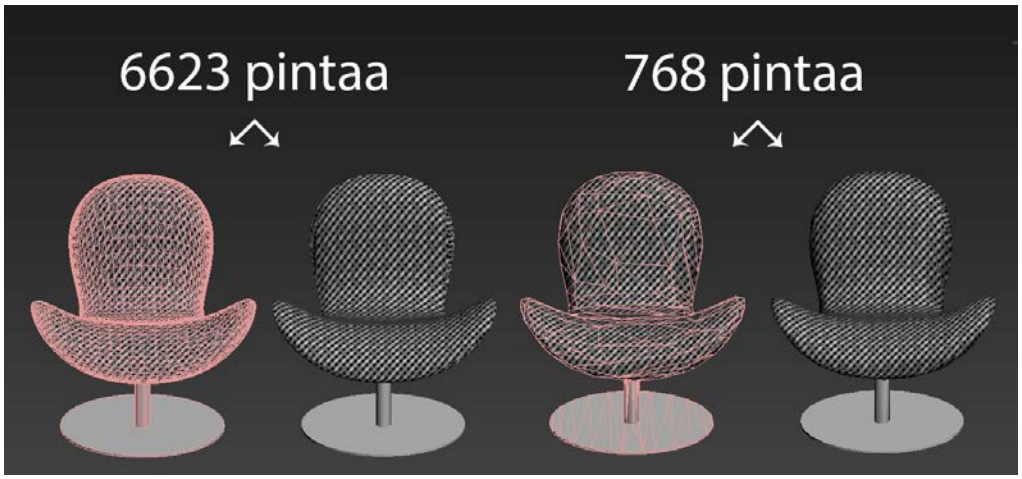


Kuva 10. 3D-malli rakennuksen kerroksesta, jonka rakenne koostuu tasaisista pinnoista ja suorista kulmista. 3D-mallissa on 38 osaa, joissa on yhteensä 8 527 pintaa.

Vaikka CAD-sovelluksissa on tehokkaat ja monipuoliset työkalut rakennusten tekemiseen, 3D-mallin rakenne ei ole aivan optimaalinen 3D-peliympäristöön. CAD-sovellukset nimittäin tekevät 3D-malliin turhia pintoja, jotka jäävät piiloon muiden pintojen väliin tai alle. Monet ylimääräisistä pinnoista täytyy poistaa tällöin manuaalisesti, sillä ohjelmistot eivät voi tietää, mitkä pinnat ovat tarpeellisia. Optimointiin kuluva aika on tapauskohtaista riippuen rakennuksen 3D-mallin pintojen määrästä ja yksityiskohdista. Joissakin tapauksissa voi olla hankala arvioida, meneekö aikaa enemmän 3D-mallin optimointiin vai uudelleen mallintamiseen.

3D-mallinnusohjelmistoissa on kyllä 3D-mallin optimointityökaluja, mutta niiden automatiikka ei aina tuota laadukasta ja hyvin optimoitua lopputulosta. Optimointityökalu voi hävittää 3D-mallista tärkeitä muotoja, tai se voi tehdä 3D-mallin osista epäsymmetrisiä. Optimointityökalu voi myös rikkoa UV-kartoituksen, minkä voi joutua korjaamaan optimoinnin jälkeen.

Huonekaluvalmistaja Martelan internetsivuilta saa ladattua 3D-malleja yrityksen huonekaluista. Martelan Fly Me -nojatuolin 3D-mallissa on 6 623 pintaa, mikä ei ole optimaalinen tarkkuus 3D-peliympäristöön. 3D-mallin pintojen määrää vähennettiinkin 3ds Max -mallinnusohjelmiston ProOptimizer-työkalulla. 3D-mallista vähennettiin pintoja 90 %, minkä jälkeen 3D-malli koostui 768 pinnasta (ks. kuva 11). Optimointi vähensi 3D-mallin pintoja huomattavasti, mutta säilytti kuitenkin tuolin muodot. Tässä tapauksessa optimointityökalu toimi siis odotetusti. Optimoinnin tuloksena tuolin 3D-malli sisältää tarpeeksi vähän pintoja, ja näin ollen sitä on mahdollista käyttää 3D-peliympäristössä.



Kuva 11. 3D-mallin optimoinnin tulos 3ds Max -mallinnusohjelmiston ProOptimizer-työkalulla.

3.2 Tuotanto 3D-mallinnusohjelmistolla

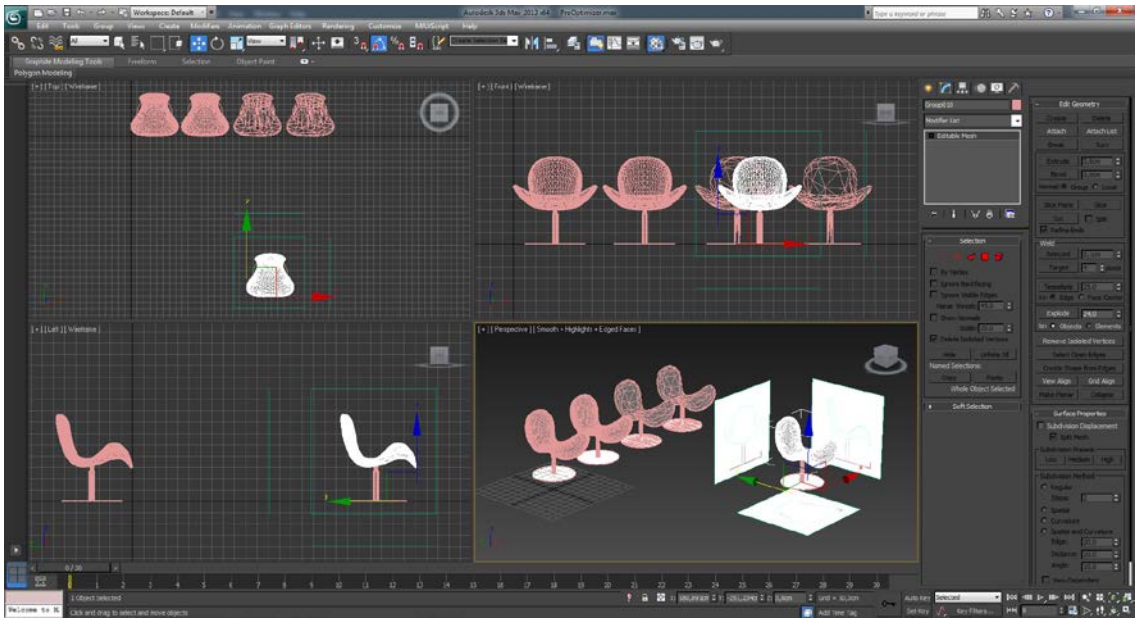
3D-mallien tekemiseen on kehitetty lukuisia 3D-mallinnusohjelmistoja. Markkinoilla on monia erilaisia 3D-mallinnusohjelmistoja, joiden käyttötarkoitus ja ominaisuudet vaihtelevat. Tarkkoihin ja realistisiin käyttötarkoituksiin tuotetuissa 3D-malleissa voi olla paljon pintoja ja yksityiskohtia, mutta 3D-peliympäristöön tarkoitettujen 3D-mallien täytyy olla kevyitä ja optimoituja. Erilaisilla 3D-mallinnusohjelmistoilla ja työkaluilla voidaan tuottaa muodoltaan samanlaisia 3D-malleja, mutta fyysinen rakenne ei aina ole samanlainen. Tämä on hyvä ottaa huomioon 3D-mallien jatkokäsittelyä silmälläpitäen.

3D-mallinnusohjelmistojen opettelun ja kokemuksen myötä voi oppia tuottamaan laadukasta 3D-sisältöä tehokkaasti moniin eri käyttötarkoituksiin (Lehtovirta & Nuutinen 2000: 22–23). 3D-mallinnusohjelmistoja, joilla pystytään tuottamaan 3D-malleja 3D-peliympäristöihin, ovat

- Autodesk Maya
- Autodesk 3ds Max
- Blender
- Autodesk Softimage XSI
- Luxology modo
- Newtek LightWave
- Maxon Cinema 4D (Software for Game Environment Artist 2013).

Ammattikäyttöön tarkoitetut 3D-mallinnusohjelmistot voivat maksaa tuhansia euroja (Autodesk store 2014). Blender on ilmainen avoimen lähdekoodin 3D-mallinnus- ja animointiohjelmisto, joka soveltuu henkilökohtaiseen käyttöön ja myös kaupalliseen käyttöön pienille yrityksille. Blenderiä kehittävät sadat vapaaehtoiset ympäri maailman. (Blender. About 2014.)

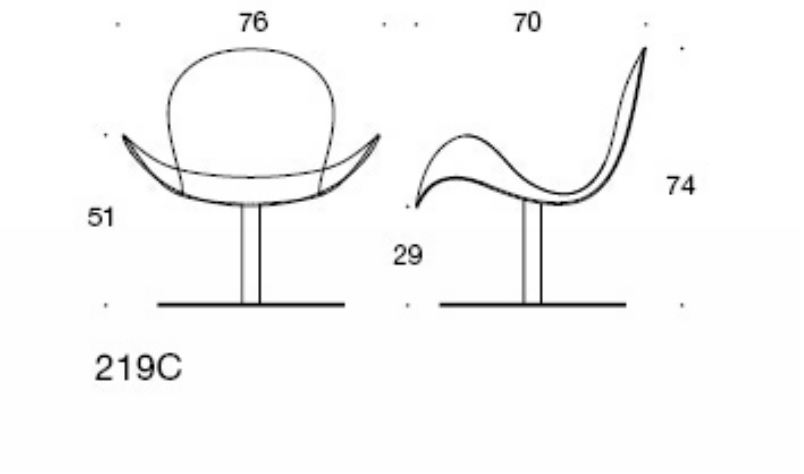
Insinöörityössä keskitytään vertailemaan 3ds Max 3D -mallinnusohjelmistoa ja sen työkaluja muihin 3D-mallinnusmenetelmiin ja sovelluksiin. 3ds Max-mallinnusohjelmisto tarjoaa tehokkaita työkaluja ja ominaisuuksia 3D-sisällön tuottamiseen pelinkehittäjille (ks. kuva 12). (3ds Max 3D-mallinnus- ja animointiohjelmisto 2014.)



Kuva 12. 3ds Max -mallinnusohjelmiston käyttöliittymä.

3D-mallinnusohjelmalla 3D-mallista on mahdollista saada optimoitu ja laadukas, koska mallia pystytään rakentamaan ja muokkaamaan tehokkaasti. UV-kartoitus ja teksturointi onnistuvat myös tehokkaasti laadukkaalla 3D-mallinnusohjelmistolla. 3D-mallinnusohjelmistoilla mallinnettaessa käytettävät pohja-aineistot ja ohjeistusmateriaalit vaikuttavat 3D-mallin tarkkuuteen ja mahdollisesti myös työskentelyaikaan. Selkeillä ohjeilla ja aineistoilla voi olla myös helpompi ja mukavampi työskennellä.

Hyvän ja tarkan viivapiirustuksen avulla voi tuottaa tehokkaasti 3D-mallin. Viivapiirustuksia on mahdollista tuoda 3D-mallinnusohjelmistoon vaikka bittikarttakuvana tai CAD-tiedostona (Harper 2012: 194; 292). Viivapiirroksia olisi hyvä olla eri puolilta objektia, ja läpileikkaukset auttavat myös hahmottamisessa (ks. kuva 13).



Kuva 13. Viivapiirros Martelan Fly Me -nojatuolista, jonka avulla voi hahmottaa objektin muodot ja mitat (Martela Fly Me 2013).

Yksi merkittävimmistä asioista 3D-mallinnusohjelmistoa käytettäessä on tekijän ammattitaito, joka vaikuttaa suoraan työn laatuun, työskentelyaikaan ja tehokkuuteen. Ohjelmiston hyvät ominaisuudet ja työkalut eivät välttämättä paranna 3D-mallin laatua tai mallinnukseen kuluvaa aikaa, mikäli tekijä ei osaa käyttää ohjelman toimintoja tehokkaasti tai jos 3D-hahmotuskyky on huono. Joillain ihmisillä voi olla ongelmia 3D-tilan ja geometrian hahmottamisessa. Kuten monessa muussakin asiassa, osalta ihmisistä 3D-mallinnus sujuu luonnostaan paremmin, koska heidän hahmotuskykynsä on parempi kuin toisilla. Eräs 3D-mallinnuksen opettaja on kuitenkin sanonut osuvasti: ”Loppujen lopuksi 3D-mallintaminen on yleensä vain pisteiden siirtelyä paikasta toiseen.”

3D-mallinnusohjelmalla mallinnettaessa objektin koolla ei ole sinänsä merkitystä, mutta yksityiskohdat ja monimutkaiset muodot voivat olla tekijälle haasteellisia, tai niiden mallintaminen voi viedä paljon aikaa. 3D-mallinnusohjelmistoilla on eroja 3D-mallin tarkastelutyökaluissa ja piirtotehokkuudessa, mikä voi vaikuttaa korkean polygonimääräisten ja monimutkaisten 3D-mallien tarkasteluun ja työstämiseen. Yleensä peliympäristöön tarkoitettujen 3D-mallien työstämisessä ei pitäisi olla ongelmia liiallisen geometrian kanssa, koska 3D-mallien tulisi olla polygonimäärältään kevyitä. Erilaiset alustat ja laitteet kykenevät piirtämään tietyn määrän 3D-geometriaa, kunnes piirtonopeus alkaa hidastua. Etenkin tuotettaessa 3D-sisältöä mobiililaitteille olisi tärkeää ottaa selvää tai kokeilla, kuinka suuri polygonimäärä alkaa hidastaa piirtonopeutta. Polygonien määrä ei ole kuitenkaan suinkaan ainoa asia, mikä vaikuttaa pelin tai 3D-sovelluksen piirtonopeuteen, mutta 3D-mallintajan kannattaa ottaa asia huomioon.

Lopuksi, kun malli halutaan saada pelimoottoriohjelmiston käyttöön, se tallennetaan 3D-mallinnusohjelmasta tiedostoformaatissa, jota pelimoottori tukee. Insinöörityössä käytetään esimerkkinä Unity3D-pelimoottoria, joka tukee seuraavia tiedostoformaatteja:

- fbx
- obj
- dae (collada)
- 3ds
- dxf.

Konversion avulla Unity3D tukee myös seuraavien 3D-mallinnusohjelmien tiedostoformaatteja:

- Max
- Maya
- Blender
- Cinema4D
- Modo
- Lightwave
- Cheetah3D (Unity3D Documentation, 3D formats 2013).

3.3 3D-malli valokuvien avulla

On olemassa ohjelmistoja, joilla pystyy tekemään 3D-mallin käyttämällä hyväksi valokuvia. 3D-mallintaminen voidaan tehdä manuaalisesti tai myös automatiikkaa hyväksikäyttäen. Joissakin ohjelmissa 3D-malli pystytään generoimaan automaattisesti käyttäjän antamia apupisteitä avuksi käyttäen, mikä nopeuttaa työskentelyä. (PhotoModeler Scanner 2014.)

Valokuvapohjaisten 3D-mallinnusohjelmistojen etuna on mahdollisesti tehokkaampi 3D-mallinnusmenetelmä ja kuvista saadut tekstuurit. Insinöörityössä tutkittiin ohjelmistoja, joilla pystytään mallintamaan valokuvista. Tavoitteena oli tutkia, löytyykö edullista

ohjelmistoa, jolla pystyisi nopeuttamaan ja helpottamaan 3D-mallinnuksen tuotantoprosessia. Valokuvapohjaisista 3D-mallinnusohjelmistoista oli aikaisempaa kokemusta ja tietoa, mutta selvitystyöhön kuului uusien tekniikoiden ja automatiikan tutkiminen. Realviz imagemodeler -ohjelmalla oli aikaisemmin tuotettu 3D-malleja, mutta tuotantomenetelmä on hidas. Ohjelmassa ei ollut automatiikkaa 3D-mallinnuksen avuksi, joten 3D-malli täytyi piirtää täysin manuaalisesti valokuviin (ks. kuva 14). Tämä tekniikka tuntui hitaalta, mutta mallista on mahdollisuus tehdä geometrisesti laadukas ja optimoitu.

Kuva 14. REALVIZ Image Modeler -ohjelma, jossa 3D-malli on piirretty manuaalisesti valokuvien avulla (Realviz Releases ImageModeler 3.5 for Windows 2003).

Kehittyneimmissä valokuviin perustuvissa 3D-mallinnusohjelmissa automatiikka hoitaa osan 3D-mallintamisesta tai jopa kokonaan. Tutkinnan kohteena olivat Eos Systemsin PhotoModeler- ja Agisoftin PhotoScan -ohjelmat, jotka on kehitetty tuottamaan 3D-malleja valokuvista. Kummankin ohjelman verkkosivuilla mainostetaan, että 3D-mallien tuottamisen pitäisi onnistua automaattisesti ja tarkasti. Kokeiluissa kummankaan valmistajan ilmaisohjelmilla ei pystynyt kuitenkaan tuottamaan helposti laadukkaita 3D-malleja. Tuhansia euroja maksavat kaupalliset lisenssit voivat mahdollisesti pystyä parempaan 3D-mallinnuslopputulokseen, mutta korkean hinnan vuoksi niitä ei pystytty hankkimaan. Automaatiotekniikan huonona puolena valokuvista mallinnettaessa voi olla mahdollinen epätarkkuus 3D-mallissa. Tehokasta ja laadukasta valokuvapohjaista

3D-mallinnusohjelmaa ei siis löydetty tässä tutkimuksessa. Uusilla tutkimuksilla ja kokeiluilla voisi kuitenkin olla mahdollista löytää tarpeeksi toimiva ohjelma, tai jos käyttöön pystyttäisiin saamaan kaupallisia lisenssejä. Tulevaisuudessa ohjelmistot voivat myös kehittyä ja halventua, mikä helpottaisi lisätutkimusta.

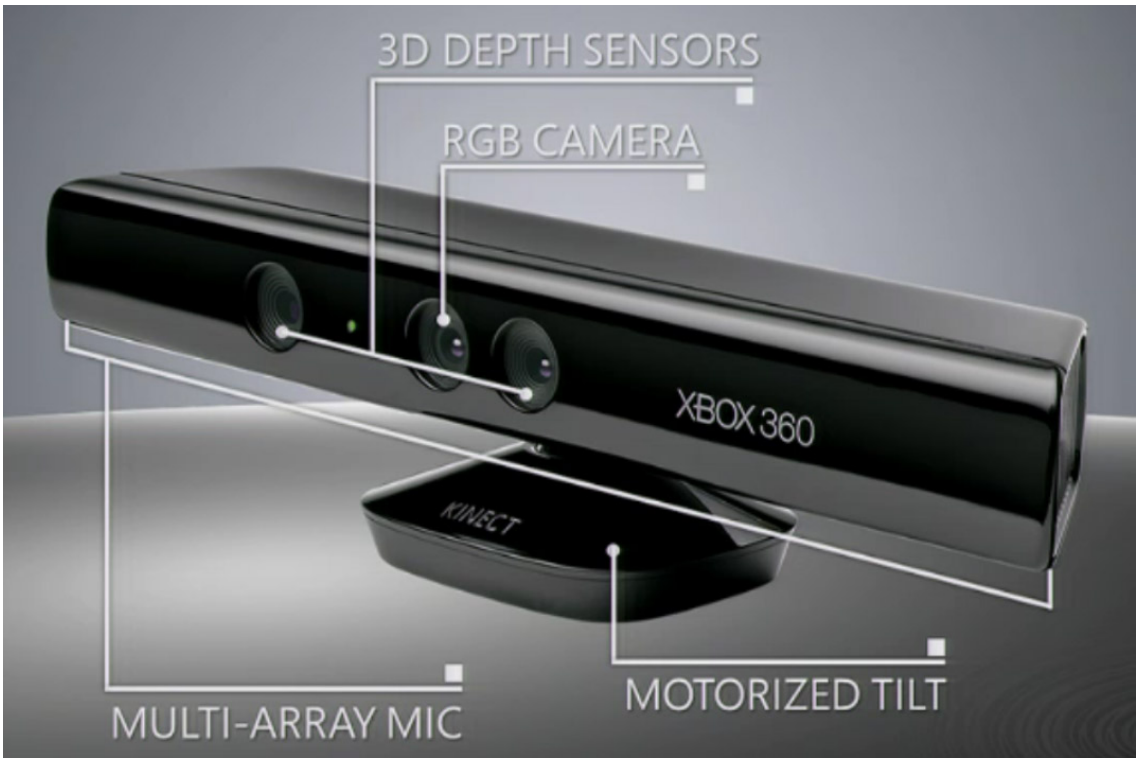
3.4 3D-skannaus

3D-skannauksen toimivuutta ja ratkaisuja tutkittiin 3D-mallien sisällöntuotannon tehostamiseksi. Tavoitteena oli löytää edullinen ja toimiva tapa skannata objekteja 3D-malleiksi. Internetistä etsittiin erilaisia 3D-skannauslaitteita ja -ohjelmistoja, joilla olisi mahdollista onnistua tavoitteessa. Hyvä ja helppo työnkulku oli myös yksi tärkeä kriteeri osana tehokasta 3D-skannauksen tuotantomenetelmää.

3D-skannauslaite ja ohjelma

3D-skannauslaitteeksi valikoitui Microsoftin Xbox 360 -pelikonsolia varten kehitetty Kinect-liikeohjain, koska se oli helposti saatavilla ja se on myös edullinen. Microsoft on kehittänyt Kinect-liikeohjainta varten ohjelmistokehitystyökalun Windows-käyttöjärjestelmiin. Ohjelmistokehitystyökalulla voidaan kehittää ohjelmia, jotka pystyvät käyttämään ja ohjaamaan Kinect-liikeohjainta (Kinect for Windows 2013).

Microsoftin Kinect-liikeohjaimessa (ks. kuva 15) on sisäänrakennettu syvyystunnistin, RGB-kamera, mikrofoni ja ihmisen luiden seuranta. Syvyystunnistin toimii infrapunasensorin avulla. Infrapunasensorissa on infrapunalähetin ja sensori, joka pystyy tunnistamaan infrapunavalosta syvyyden. Liiketunnistimen tunnistusalue on vakiona 800–4000 mm ja lähiasetuksilla 500–3000 mm. RGB-kamera pystyy kuvaamaan 1280 x 960 pikselin tarkkuudella 12 kuvaa sekunnissa tai 640 x 480 pikselin tarkuudella 30 kuvaa sekunnissa. (Kinect for Windows features 2014.)



Kuva 15. Microsoftin Kinect -liikeohjain ja sen ominaisuudet (Wave hello to Microsoft's Kinect 2010).

Internetistä etsittiin 3D-skannausohjelmia, jotka pystyivät käyttämään Kinect-liikeohjainta. Joissakin testatuissa 3D-skannausohjelmissa oli ongelmia 3D-mallin tarkkuudessa tai ohjelman toimivuudessa. Jotkin ohjelmat rakensivat 3D-mallia reaaliaikaisesti, kun 3D-skannausta suoritettiin. Eräs ohjelma rakensi 3D-mallia otos kerrallaan. Ohjelma lisäsi 3D-datan aina otoksen jälkeen. Ongelmia oli kuitenkin 3D-datan yhdistämisessä, kun uusi otos ei aina asettunut oikeaan kohtaan. Monissa ohjelmissa oli ongelmana 3D-mallin heikko laatu, ominaisuudet tai tallennusformaatit olivat huonoja.

3D-skannausohjelmaksi valittiin ohjelma nimeltä Skanect. Se tuotti testien laadukkaimmat 3D-mallit. Skanect todettiin helppokäyttöiseksi, ja siinä oli hyvät työkalut. Ohjelma generoi 3D-malliin skannauksen ohessa myös tekstuurin, jonka pystyi tallentamaan bittikarttakuvaksi. 3D-mallin pystyi tallentamaan tiedostoformaateissa, joita 3D-mallinnusohjelmistot tukevat. Skanect Pro -ohjelma 3D-skannaa reaaliajassa ja rakentaa teksturoidun 3D-mallin minuuteissa. Lisäksi Skanect Pro -versio maksaa vain 99 € (Get Skanect 2014).

3D-skannauksen työnkulku

Kun Skanect-ohjelma ja Kinectiin tarvittavat ajurit on asennettu, voi 3D-skannauksen aloittaa. Jos skannattava objekti tai alue oli suuri, käytettiin Kinectissä pitkää usb-kaapelia ja kannettavaa tietokonetta, joita oli mahdollista siirtää skannauksen aikana. Skannattavan objektin ympärille kannatti varata liikkumatilaa, sillä Kinect tarvitsi joissain tapauksissa paljon etäisyyttä skannattavasta kohteesta. On myös huomioitava, että skannattavan objektin lähellä ei tulisi olla muita objekteja, jotka voivat häiritä 3D-skannausta.

Skanect-ohjelmassa työnkulku todettiin helpoksi ja lineaariseksi. Työnkulussa on viisi vaihetta, joiden mukaan etenemällä 3D-skannaus on selkeää:

- valmistelu
- nauhoitus
- rekonstruointi
- käsittely
- tallentaminen ja jakaminen.

Ohjelman graafinen ulkoasu ja käyttöliittymä olivat selkeät. Ensimmäisessä valmisteluvälilehdessä valitaan skannattavan objektin tai alueen koko ja tallennuskansio (ks. kuva 16). Välilehdessä on myös mahdollista ladata tallenne, saada lisenssitiedot ja muokata asetuksia.

Kuva 16. Skanect-ohjelman ensimmäinen vaihe, jossa määritellään 3D-skannattavan alueen koko ja tiedoston tallennuspolku.

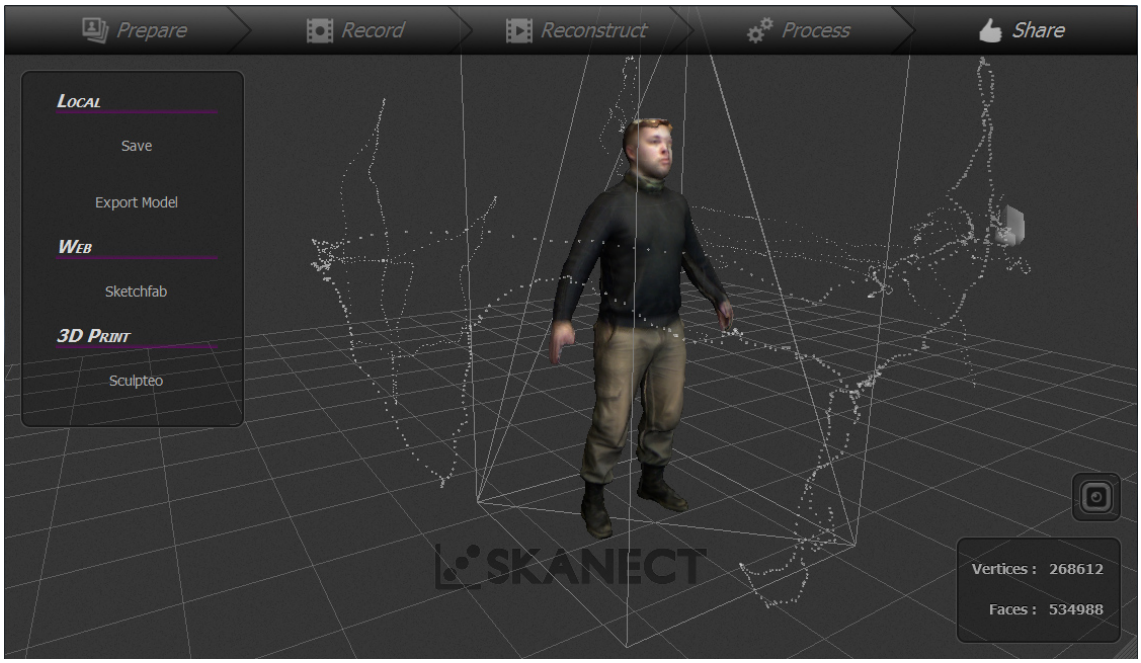
Seuraavassa nauhoitusvälilehdessä voidaan aloittaa itse 3D-skannaaminen. Ennen 3D-skannauksen aloittamista oli mahdollista rajoittaa 3D-mallin polygonien määrää tai asettaa nauhoittamiselle ajastus. Ruudulla näkyi monta tarpeellista ja mielenkiintoista ikkunaa, joista näki Kinectistä saapuvaa dataa ja kuvaa (ks. kuva 17). Piirtyvän 3D-mallin ympärillä oli kolmiulotteinen tila, jossa viivamainen laatikko esitti skannattavan alueen rajoitukset, ja laatikon sisällä näkyi Kinectistä tuleva reaaliaikainen 3D-malli.

Kuva 17. Skanect-ohjelman 3D-skannauksessa Kinectistä saapuvaa dataa ja kuvaa.

Ennen nauhoituksen aloittamista Kinect kannatti sijoittaa mahdollisimman suoraan objektia kohti ja tarkistaa näytöltä, että objekti on oikealla korkeudella kolmiulotteisessa tilassa. Objektin 3D-skannauksessa Kinectiä kannatti viedä hiljalleen objektin ympäri ja yrittää osoittaa sitä objektin kaikkiin kohtiin, mikä vähensi aukkojen jäämistä 3D-malliin. Jos Kinectin vei liian lähelle objektia tai jos ohjelma ei löydä tarpeeksi ulottuvuuksia objektista, voi ohjelma hävittää 3D-mallin sijainnin 3D-avaruudessa. Virheen voi yrittää korjata viemällä Kinectin tarkalleen samaan kohtaan, jossa ohjelma kadotti kalibroinnin ja kokeilla, löytääkö ohjelma oikean kalibroinnin. Eräs helpompi tapa voisi olla asettaa 3D-skannattava objekti pyörivälle alustalla, jolloin Kinectiä ei tarvitsisi liikutella objektin ympärillä. Pyörivän objektin 3D-skannaus voisi olla tarkka ja hyvä tapa 3D-skannaukseen.

Kun 3D-skannaus on suoritettu, se on mahdollista rekonstruoida, jotta lopputuloksesta tulisi mahdollisimman hyvä. Rekonstruktiovaiheessa ohjelma käy 3D-skannauksen tarkemmin läpi ja rakentaa tarkemman 3D-mallin kuin reaaliajassa suoritettu mallinnus. Kun 3D-skannaus on rekonstruktoitu, siirrytään käsittelyvaiheeseen. Käsittelyvaiheessa oli mahdollisuus käyttää työkaluja, joilla pystyi muokkaamaan ja korjaamaan 3D-mallia.

Viimeinen vaihe Skanect-ohjelmassa oli 3D-mallin tallentaminen ja jakaminen. 3D-mallin pystyi tallentamaan Skanect-ohjelmasta PLY-, OBJ-, STL- tai VRML-tiedostoformaatteihin. 3D-malli oli myös mahdollista tallentaa Sketchfab- tai Sculpteo-pilvipalveluihin, joissa 3D-mallia pystyy tarkastelemaan selaimessa ja jopa lähettää tiedosto 3D-tulostettavaksi (ks. kuva 18).



Kuva 18. Skanect-ohjelman viimeinen vaihe, jossa 3D-mallin voi tallentaa erilaisiin tiedostomuotoihin tai ladata pilvipalveluihin.

Kinect-liikeohjain ja Skanect-ohjelma luovat yhdessä kohtuullisen hyvän 3D-mallin teksturoituna, mutta parantamisen varaa on vielä 3D-mallin ja tekstuurin tarkkuudessa. 3D-skannatusta ihmishahmon 3D-mallista kuitenkin pystyi huomaamaan, että pienet muodot eivät tulleet tarkasti 3D-malliin. Erilaisissa 3D-skannausohjelmissa pystyi huomaamaan, ettei Kinect-liikeohjain ole kovinkaan tarkka alle 10 cm:n kokoisissa muodoissa. Skanect-ohjelman generoima tekstuuri 3D-mallissa oli myös heikkolaatuinen, koska tekstuurin luontitekniikka ei ollut tarpeeksi toimiva. Ohjelma antaa 3D-mallin jokaiselle pinnalle oman värin, joista se voi generoida tekstuurin bittikarttakuvaksi tallennuksen yhteydessä. Ongelma on kuitenkin siinä, että 3D-mallissa voi olla 3D-skannauksen jälkeen jopa 300 000 pintaa, joten 3D-mallia täytyy optimoida. Ohjelmassa pystyy sisäisesti vähentämään pintojen määrää, mutta tällöin tekstuurin tarkkuus ja laatu huonontuvat (ks. kuva 19).

Kuva 19. Skanect-ohjelmalla 3D-skannatun 3D-mallin optimoinnista seuraa tekstuurin huononTuminen, kun 3D-mallin pintojen määrää vähennetään.

Ongelma ratkaistiin osittain valokuvaamalla 3D-skannattu objekti ja tekemällä optimoitu ja selkeä tekstuuri kuvankäsittelyohjelmassa. Ratkaisun takia tuotantomenetelmän prosessiin jouduttiin lisäämään työtehtäviä, mutta tällä ratkaisulla 3D-mallin tekstuurista sai paremman. 3D-mallin optimointi oli myös mahdollista toteuttaa 3D-mallinnusohjelmistossa. 3ds Max -ohjelman ProOptimizer-työkalulla 3D-mallia pystyttiin optimoimaan ja jäljelle jäi noin 3 000 pintaa (ks. kuva 20). Lisäksi 3D-mallinnusohjelmassa täytyi tehdä 3D-mallille uusi UV-kartta, jotta kuvankäsittelyohjelmassa tehdyn tekstuurin sai hyvin 3D-malliin.

Kuva 20. 3D-skannatun 3D-mallin optimoinnin tulos ja kuvankäsittelyohjelmassa tehty tekstuuri.

3D-skannauksen työnkulun yhteenvetona voidaan todeta, että edullisilla ratkaisuilla on hankala saada tuotettua laadukasta 3D-mallia 3D-peliympäristöön. Monia laitteita ja ohjelmistoja on tarjoilla, mutta edullista ratkaisua etsittäessä on hankala päästä hyvään ja käyttökelpoiseen lopputulokseen. Edullisilla 3D-skannausmenetelmillä ei ole kannattavaa tuottaa 3D-malleja 3D-peliympäristöön, mutta 3D-malleille voi löytyä käyttöä johonkin muihin käyttötarkoituksiin. Markkinoilta löytyy paljon erilaisia 3D-skannereita, joiden laatu, hinta ja tekniikka vaihtelevat. Monet 3D-skannerit ovat arvokkaita, ja niistä on vaikea tietää ilman kokeilua, tuottavatko ne laadukkaita 3D-malleja ja tekstuureja.

4 360°-panoraamakuvien tuottaminen ympäristöstä

360°-panoraamakuva on kuva, joka sisältää 360° näkymän ympäristöstä (Harper 2012: 622). 360°-panoraamakuvien tuotantomenetelmä oli keskeinen osa insinöörityötä, ja prosessin kehittäminen oli yksi tärkeimmistä tehtävistä. Tavoitteena oli luoda tuotantoprosessi 360°-panoraamakuvien tuottamiseen, jota voi käyttää 3D-peliympäristöissä. 3D-pelimoottoreissa ja muissa 3D-sovelluksissa panoraamakuvia voi käyttää hyväksi esimerkiksi materiaalien heijastuksissa ja ympäristön visualisoimisessa. Prosessin kehittäminen oli tärkeää, koska virtuaalisten rakennusten ulkopuolelle haluttiin saada maisema aidosta ympäristöstä.

4.1 Kuvausmenetelmät ja kuvienkäsittely

360°-panoraamakuvia pystyy tekemään aidoista ympäristöistä erilaisilla valokuvaustekniikoilla ja -ohjelmistoilla. Ympäristöstä on mahdollista valokuvata monta kuvaa ja yhdistää ne panoraamakuvaksi, kuten kuva 21 hyvin havainnollistaa. Kuvaaminen voi tapahtua pitelemällä kameraa käsin tai esimerkiksi jalustalla. Kuvattaessa useita valokuvia yhdellä kameralla olisi tärkeää pystyä pitämään kamera samassa pisteessä koko kuvauksen ajan. Jos valokuviin syntyy perspektiivimuutoksia, on valokuvia vaikeampi yhdistää saumattomasti panoraamakuvaksi. Kokeiluissa on huomattu, että jotkin ohjelmat ja sovellukset voivat yhdistää valokuvia panoraamakuvaksi paremmin kuin toiset. Myös perspektiivi- ja valotusmuutosten korjaaminen panoraamakuviin onnistuu paremmin laadukkaimmilla ohjelmilla.

Kuva 21. Yläpuolella näkyvistä kuudesta valokuvasta on yhdistetty alempi panoraamakuva.

Panoraamakuvia tehtäessä on huomattu, että kaikkien valokuvien olisi hyvä sisältää noin puolet kuvattavan kohteen ympäriltä otettujen muiden kuvien informaatiosta. Siten kuvien yhdistelemisessä olisi enemmän säätövaraa ja kuvasta voisi saada mahdollisimman saumattoman. Kehittyneimmät ohjelmat pystyvät yhdistämään valokuvia automaattisesti ja tekemään saumatonta jälkeä, kun taas tarkoitukseen kehitetyillä objektiiveilla saa panoraamakuvat yhdestä valokuvasta.

Nykyään useissa digitaalikameroissa on panoraamakuvausominaisuus sisäänrakennettuna, ja uusille älypuhelimille pystyy asentamaan panoraamakuvaussovelluksia. Nokia 1020 -älypuhelimella on testattu Photosynth-ohjelmaa (ks. kuva 22), joka on panoraamasovellus Windows Phone -käyttöjärjestelmissä. Ohjelmalla pystytään tekemään 360°-panoraamakuvia, mutta testeissä panoraamakuvista oli hankala saada virheettömiä. Myös tietokoneille on kehitetty paljon ohjelmia, joilla pystyy tekemään panoraamakuvia valokuvista. 360°-panoraamakuvien tuotantomenetelmän kehityksen aikana on testattu useita tietokoneille kehitettyjä ohjelmia, joilla voidaan tehdä 360°-panoraamakuvia. Useilla testaajilla ohjelma ei kuitenkaan pystynyt tekemään edes kohtalaisen saumattomia 360°-panoraamakuvia.

Kuva 22. Nokian 1020-älypuhelimen Windows Phone -käyttöjärjestelmässä toimivalla Photosynth-mobiilisovelluksella toteutettu panoraamakuva.

Ammattikäyttöön kehitetyillä monikamerajärjestelmillä (ks. kuva 23) tai 360°-panoraamaobjektiiveilla pystyy tuottamaan 360°-panoraamakuvia nopeammin ja laadukkaammin kuin yhdistelemällä yhdellä kameralla otettuja valokuvia. Monikamerajärjestelmissä kamerat on yhdistetty toisiinsa, ja ne ottavat kuvan samaan aikaan, joten kuvaajan tekemiä perspektiivivirheitä on vähemmän. Kamerat on asetettu niin, että ne ottavat valokuvat ympäristön joka kohdasta. Joihinkin monikamerajärjestelmiin on integroitu ohjelma, joka tekee automaattisesti valokuvista 360°-panoraamakuvan.

Kuva 23. GoPro Hero 3 kameroista rakennettu monikamerajärjestelmä 360°-panoraamakuvaukseen (F360 Explorer 2014).

Panoraamakuvia on myös mahdollista kuvata tarkoitukseen kehitetyllä objektiivilla. Panoraamaobjektiivia käytettäessä panoraamakuva syntyy suoraan valokuvattaessa eikä kuvia tarvitse yhdistellä. Panoraamaobjektiivilla panoraamakuvat ovat myös virheettömämpiä, koska kuvien yhdistämisessä syntyy helposti pieniä virheitä kuvaan.

Kuva 24. Järjestelmäkameraan asetettava 360°-panoraamaobjektiivi ja objektiivin avulla syntyvä 360°-panoraamakuva (360 Virtual Tour Cameras 2014).

Käsin tai jalustalla kuvaamisessa on kuitenkin omat rajoituksensa, mikäli kuvauspaikkaan on hankala päästä tai jos valokuva täytyy kuvata ilmasta. Joissakin tapauksissa käyttöön onkin otettava luovempia ratkaisuja. Ongelman ratkaisu voi olla radio-ohjattava kopteri, johon pystytään liittämään digitaalikamera. Nykyään markkinoilla on monenlaisia radio-ohjattavia koptereita, joihin pystyy kiinnittämään erilaisia digitaalikameroita. Joissakin valokuvaukseen kehitetyissä koptereissa on vähintään neljä roottoria, mutta toisissa jopa kahdeksan. Kopterin kantokyky ja liitäntämahdollisuudet määrittävät pitkälti sen, minkälaisen kameran kopteriin voi yhdistää. Joissakin koptereissa on valmiiksi integroituna kamera, mutta se ei välttämättä ole laadukas. Laadukkain ratkaisu on kiinnittää järjestelmäkamera kopteriin, mikäli kopteri vain pystyy lentämään sen kanssa. Kopterit, joihin saa liitettyä vaikka suositun GoPro Hero -digitaalikameran, maksavat alle 500 €. Ammattilaiskäyttöön tarkoitetut kopterit, jotka jaksavat kantaa järjestelmäkameran, voivat maksaa jopa tuhansia euroja.

4.2 360°-panoraamakuvien käyttö 3D-peliympäristössä

3D-pelimoottoreissa ympäristön kuvaamisessa ja esittämisessä voidaan käyttää erilaisia tekniikoita. Spheremap ja cubemap ovat projisointitekniikoita, joita käytetään peliteknologiassa ympäristön esittämisessä ja materiaalien heijastuksissa. Muita projisoin-

titekniikoita ja nimityksiä ovat myös skybox ja sky dome, jotka toimivat samoilla periaatteilla. (Ahearn 2006.)

Spheremap- ja cubemap-tekstuurit eroavat kuitenkin toisistaan paljon. Spheremap on tarkoitettu projisoitavaksi pallomaiseen muotoon, jolloin sitä voi käyttää parhaiten ympyrän muotoisissa objekteissa (ks. kuva 25). Kun ympäristöä halutaan kuvata spheremapin avulla, käytetään usein suurta pallon muotoista objektia, johon spheremap-tekstuuri asetetaan. Kun spheremap-objektin sisällä katselee ympärilleen, näyttää siltä, että ympärillä oleva tausta näkyy joka suunnasta. Esimerkiksi jos liikkuu kolmiulotteisessa pelissä rakennuksessa ja katsoo ulos ikkunasta, spheremapin avulla rakennuksen ympäristö näyttää erilaiselta joka suunnasta. Jos spheremap-objekti on oikean kokoinen suhteessa kameran liikkeen nopeuteen, voi objektista nähdä perspektiivin muutoksen.

Kuva 25. 360°-panoraamakuva, jota voi käyttää spheremap-projisoinnissa.

Skybox on tekniseltä toteutukseltaan samantyyppinen spheremapin kanssa, mutta se projisoidaan laatikon muotoiseen objektiin. Skyboxia voi kutsua myös nimellä cubemap. Skyboxia esittävässä laatikossa on kuusi pintaa, joihin tulee eri tekstuuri (ks. kuva 26). Skyboxia käytetään yleensä esittämään vain todella kaukaista aluetta, sillä jos skybox näyttää siirtyvän kameran liikkuessa, perspektiivinmuutos voi näyttää epärealistiselta. Onkin huomattu, että skybox on 3D-pelimoottoreissa yleisimmin käytetty taustankuvaamistekniikka. Unity3D-ohjelmassa skybox-ominaisuus on sisäänrakennettu

pelimoottoriin ja skybox esittää ympäristön taustaa, mutta sitä voi myös käyttää hyväksi materiaaleissa.

Kuva 26. Kuusi kuvaa, jotka ovat toistensa kanssa saumattomia. Kuvia voi käyttää Unity3D-pelimoottorissa cubemap-materiaalissa.

4.3 Tuotantoprosessi

Tuotantomenetelmän prosessissa oli tarkoitus tuottaa 360°-panoraamakuva ja käyttää sitä 3D-peliympäristössä. 360°-panoraamakuvien tuotantoprosesseihin kuului löytää oikeat kuvausmenetelmät, tehokas keino yhdistää kuvat 360°-panoraamakuvaksi ja hyvän käyttötavan löytäminen 3D-peliympäristössä. 360°-panoraamakuvan haluttiin esittävän 3D-peliympäristössä virtuaalisen toimistorakennuksen ulkopuolista maisemaa. Tavoitteena oli tehdä tuotantoprosessista tehokas, laadukas ja edullinen.

Valokuvauksen menetelmän kehittämiseen vaikutti vaatimus, että kuvia on otettava mahdollisesti korkealta ilmasta. Parhaaksi vaihtoehdoksi todettiin radio-ohjattava helikopteri, johon sai liitettyä kameran. Radio-ohjattavan helikopterin avulla valokuvia voisi kuvata 360°-panoraamakuvia varten ilmasta ja muista hankalista paikoista. Erilaisia radio-ohjattavia helikoptereita vertailtiin niiden ominaisuuksien, kantokyvyn ja hinnan perusteella. Radio-ohjattavaksi helikopteriksi valittiin Dji Phantom 2 quadro -kopteri, johon oli saatavilla kiinnitys GoPro Hero 3 -kameraan. Näin löytyi kokonaisuus, johon sisältyi Dji Phantom 2 -kopteri ja GoPro Hero 3 -digitaalikamera (ks. kuva 27). Laitteet maksoivat yhteensä noin 800 €.

Kuva 27. Valokuva ilmassa lentävästä Dji Phantom 2 quadro -kopterista, johon on kiinnitetty GoPro Hero 3 -kamera.

Dji Phantom 2 oli kätevä väline ilmakuvaukseen. Kopteri oli helppo ottaa käyttöön, koska internetistä löytyi hyvät opetusvideot valmistajan verkkosivuilta. Kopteri on pyritty kehittämään helpoksi ja turvalliseksi lentää gps-paikantimen ja gyroskooppiapuvälineiden avulla. Apuvälineet auttavat pitämään kopterin automaattisesti paikoillaan ja jopa laskeutumaan itse lähtöpaikkaan, mikäli kaukosäätimessä tapahtuu virhe tai kopteri lentää liian kauas. Kopterissa on neljä roottoria, jotka pitävät lennon tasaisena, ja nostovoima riittää noin yhdelle kilolle. Lentoetäisyys on avoimella alueella noin kilometri, ja lentoajaksi luvataan noin 25 minuuttia. (Phantom 2 features 2014.)

Kopteriin liitettävä GoPro Hero 3 -kamera on pienestä koostaan huolimatta monipuolinen ja laadukkaan oloinen. Kamerassa on 12 megapikselin kenno, jolla saa 4000 x 3000 -resoluutioisia laajakuvia (ks. kuva 28). Kameran voi asettaa ottamaan kuvia tietyin aikavälein, mikä on kätevä ominaisuus 360°-panoraamakuvauksessa. Videokuvaus kykenee nauhoittamaan jopa 4K-resoluutioista kuvaa 15 kuvaa sekunnissa, mutta vielä parempi on käyttää 1080p-resoluutioista kuvaa 60 kuvaa sekunnissa, jolloin kuva on erittäin hyvälaatuista. Kameran saa myös kätevästi vedenpitävään kuoreen, mikäli valokuvaus suoritetaan huonoissa olosuhteissa. (Hero3+ black edition tehnical specs 2014.)

Kuva 28. GoPro Hero 3 -kameran laajakuva-asetuksella otettu maisemakuva ilmasta.

Kopterilla kokeiltiin erilaisia kuvaustapoja, joilla saataisiin 360°-panoraamakuviin paras mahdollinen tulos. Parhaaksi tavaksi todettiin kameran asettaminen valokuvaamaan muutaman sekunnin välein, minkä jälkeen kopterilla lennettiin haluttuun kohteeseen ja pyörittiin samassa pisteessä paikoillaan. Tällä tavalla kamera otti ympäristöstä valokuvat joka puolelta. Kahdeksan valokuvaa riitti hyvin saamaan GoPro Hero 3 -kameralla näkymän joka puolelta ympäristöä, koska linssillä saa laajoja kuvia. Yleensä valokuvia kannatti ottaa varmuuden vuoksi eri korkeuksilta, jotta 360°-panoraamakuvan näkymän korkeuden valinnassa olisi vaihtoehtoja. Maanpinnasta pystyi ottamaan valokuvia asettamalla kamera valokuvaamaan alaspäin, asettamalla ajastinlaukaisu ja lentämällä kopterilla mahdollisimman ylös. Näin kamera otti maanpinnasta valokuvia eri korkeuksilta, kuten kuvasta 29 hyvin huomataan.

Kuva 29. GoPro Hero 3 -kameralla otettu valokuva maanpinnasta. Kamera oli asetettu Dji Phantom 2 quadro -kopteriin kuvaamaan alaspäin.

Kokeiluissa oli mukana useita ohjelmia, joilla pystyi yhdistämään kuvia 360°-panoraamakuvaksi. Monissa ohjelmissa tuntui kuitenkin olevan ongelmia GoPro-kameran linssivääristymän kanssa, koska kamera ottaa hyvin laajoja kuvia ja kuvissa perspektiivi vääristyy voimakkaasti. Kuvien yhdistäminen 360°-panoraamakuvaksi onnistui kuitenkin tehokkaasti Kolorin Autopano Giga -ohjelmalla, jolla pystyy tuottamaan 360°-panoraamakuvia. Valitut kuvat ladataan ohjelmaan, joka kalibroi kuvat automaattisesti ja yhdistää ne 360°-panoraamakuvaksi. Autopano Giga -ohjelma tuntui käytössä laadukkaalta ja ominaisuuksiltaan hyvältä. Panoraamakuvan tallennuksen yhteydessä oli paljon ominaisuuksia ja säätömahdollisuuksia. Autopano Giga -ohjelmassa pystyi valitsemaan panoraamakuvan projisoinniksi spheremap-muodon, ja kuva tallennettiin 4096 x 2048 resoluution tarkkuudella PSD-tiedostoformaattiin.

Viimeisenä vaiheena pyrittiin löytämään paras tapa esittää 360°-panoraamakuva 3D-peliympäristössä. Tavoitteena oli luoda 3D-malli, johon voi asettaa 360°-panoraamakuvan. Kun 360°-panoraamakuva on kartoitettu oikein pallon muotoiseen objektiin ja objektia katselee sisältäpäin, kuva näyttää saumattomalta ja sitä voi katsoa

eri suunnista. 3ds Max -mallinnusohjelmassa mallinnettiin pallonmuotoinen objekti, johon asetettiin pallonmuotoinen UV-kartta. Objektin pinnat käännettiin sisäänpäin, ja 3D-malli tallennettiin FBX-tiedostoformaattiin. 3D-mallin vietiin Unity3D-pelimoottoriin, jossa siihen liitettiin 360°-panoraamakuvan sisältävä materiaali. 360°-panoraamakuvan materiaali täytyi asettaa kirkkaaksi niin, ettei valaistus vaikuta siihen. Näin tekstuurin kirkkaus saatiin pysymään vakiona, eikä mikään valaistus vaikuttanut taustan kirkkauteen. Lopuksi pallonmuotoisen objektin skaala säädettiin sopivaksi, jotta se näkyisi virtuaalisen rakennuksen ulkopuolella. Maantasolla olevien kerroksen näkymässä pallon muotoinen 3D-malli ei kuitenkaan näyttänyt hyvältä, koska maa näytti syvenevän rakennuksen alle. Ongelma ratkaistiin tekemällä puolipallonmuotoinen objekti, jonka puolessavälissä oli tasainen pinta. Tasainen kohta asetettiin toimistorakennuksen alle. Siten maantaso näytti jatkuvan ulkopuolella lattian perspektiivin mukaan, ja perspektiivi näytti kohtuullisen hyvältä, kuten kuvasta 30 voi hyvin havaita.

Kuva 30. Vasemmalla 360°-panoraamakuvat on asetettu pyöreisiin 3D-malleihin Unity3D -pelimoottorissa rakennuksen ympärille ja oikealla on näkymät rakennuksien sisältä, jossa 360°-panoraamakuvat esittävät taustalla ympäristöä.

5 3D-sisällöntuotanto tulevaisuudessa

Tulevaisuuden 3D-sisällöntuotannon menetelmiä on hyvä tutkia, jotta 3D-sisällöntuotannon prosesseja voisi mahdollisesti parantaa tulevaisuudessa. 3D-mallien ja myös panoraamakuvien tuottaminen näyttää tulevaisuudessa valoisalta, sillä uusia menetelmiä kehitetään helpommiksi, laadukkaammiksi ja automaattisemmiksi. 3D-skannausta ja valokuvista mallintamista kehitetään varmasti tulevaisuudessa paremmiksi. Todennäköisesti 3D-laserskannerit halpenevat, ja markkinoille tulee yhä kehittyneempiä ja parempia laitteita.

Internetissä oli taannoin uutinen, jossa kerrottiin mobiililaitteille kehitetystä 3D-skannaussovelluksesta. Mobiililaitteella olisi mahdollista pystyä tekemään 3D-malleja valo- tai videokuvaamalla (ks. kuva 31). Olisikin hienoa, mikäli objekteja pystyisi tulevaisuudessa 3D-skannamaan älypuhelimella.

Kuva 31. 3D-skannausta mobiililaitteella (3-D scanning, with your smartphone 2014).

Mallintaminen valokuvista ei ole sinänsä uusi asia, mutta tulevaisuudessa se voisi olla helpompaa, laadukkaampaa ja automatisoidumpaa. 3D-mallien optimoinnin automatisointi 3D-peliympäristöön voi olla vielä kauan käsityötä, mutta siihenkin voidaan kehittää parempia, laadukkaampia ja tehokkaampia ratkaisuja.

Panoraamakuvauksessa monikameratekniikka ja automaattinen kuvien yhdistäminen on jo pitkälle kehittynyttä, ja panoraamakuvia saa nykyään tehtyä helposti. Internetissä esiteltiin laitetta, joka ottaa valokuvat monikameratekniikalla ja luo 360°-panoraamakuvan automaattisesti. Panono on 360°-panoraamakuvauslaite (ks. kuva 32), joka on tehty pallonmuotoiseksi, ja siinä on 36 kameraa. Laitteen voi heittää pystysuoraan ilmaan, ja se ottaa valokuvat korkeimmassa kohdassa; sillä voi myös ottaa kuvia napista painamalla tai sille kehitetyllä jalustalla. Laite on varsin kätevä, koska sillä voisi ottaa 360°-panoraamakuvia helposti myös korkealta. (Panoramic ball camera, product features 2013.)

Kuva 32. Panono-360°-panoraamakuvauslaite ja sen käyttömahdollisuuksia (Panoramic ball camera, product features 2013).

6 Yhteenveto

Insinöörityössä tutkittiin ja kehitettiin erilaisia 3D-sisällöntuotannon menetelmiä 3D-peliympäristöön. Pääaiheina olivat 3D-mallien tuotantomenetelmät ja 360°-panoraamakuvien tuottaminen ympäristöstä 3D-peliympäristöön. 3D-mallien tuotantomenetelmissä tutkittiin tehokkaita ja laadukkaita tapoja tuottaa erikokoisia ja -tyyppisiä 3D-malleja 3D-peliympäristöön. Tutkimuksiin kuului kokeilla erilaisia ohjelmistoja, jotka käyttäisivät 3D-mallien tuottamiseen automaatiotekniikkaa. Ratkaisun täytyi kuitenkin olla edullinen ja tuotantoprosessia helpottava. Toisena tavoitteena oli myös kehittää toimiva prosessi 360°-panoraamakuvien tuottamiseen ympäristöstä eri korkeuksilta. 360°-panoraamakuvien tuotantoprosesseihin kuului löytää oikeat kuvausmenetelmät, tehokas keino yhdistää kuvat 360°-panoraamakuvaksi ja hyvän käyttötavan löytäminen 3D-peliympäristössä.

3D-mallien tuotantomenetelmien kehityksessä keskityttiin tutkimaan 3D-mallien tuottamista valokuvien avulla ja 3D-skannaamalla. Teoriassa erilaisilla 3D-mallinnusautomaatiotekniikoilla olisi mahdollista nopeuttaa ja parantaa 3D-sisällöntuotantoa. Suurin ongelma 3D-mallien tuottamisessa 3D-peliympäristöön on kuitenkin 3D-mallin ja tekstuurin optimointi. 3D-skannauksella voidaan tuottaa tarkka 3D-malli, mutta ongelmana on suuri pintojen määrä ja tekstuurin tuottaminen hyväksi. Kinect on edullinen ratkaisu 3D-skannaukseen, mutta saako se 3D-malleista ja tekstuureista tarpeeksi hyvät ja laadukkaat? Eräs vaihtoehto on myös 3D-mallin tekeminen valokuvien avulla. 3D-mallintaminen kalibroitujen valokuvien päälle on kuitenkin hidas tapa työskennellä, ja samantyyppistä 3D-mallinnusta voi toteuttaa 3D-mallinnusohjelmalla. Nopeammin 3D-mallin voisi saada valokuvista automaatiotekniikalla, mutta hyvään lopputulokseen pääseminen vaatii käsityötä, eikä 3D-mallista tule välttämättä kovinkaan optimoitua. Useissa tapauksissa 3D-mallia tai tekstuuria joudutaankin korjaamaan toimivaksi 3D-peliympäristössä. Markkinoilla voi olla mahdollisesti tehokkaita 3D-mallinnusratkaisuja, joilla voitaisiin tuottaa ja optimoida 3D-malleja 3D-peliympäristöön. Ratkaisuja ei kuitenkaan tunnu olevan helposti ja edullisesti saatavilla. Tehokkaimpana 3D-mallinnusratkaisuna voidaankin pitää ammattilaista, joka tekee 3D-mallit 3D-peliympäristöön 3D-mallinnusohjelmistolla.

360°-panoraamakuvien tuotantoprosessi kehitettiin onnistuneesti. Kokeilun kautta löytyivät hyvät laitteet ja ohjelmisto 360°-panoraamakuvien tuottamiseen. Valokuvauslaitteina toimivat hyvin Dji Phantom 2 quadro -kopteri ja GoPro Hero 3 –kamera. Valokuvi-

en yhdistäminen 360°-panoraamakuvaksi onnistui puolestaan parhaiten Kolorin Autopano Giga -ohjelmalla. 3D-peliympäristössä 360°-panoraamakuvaa pystyttiin käyttämään parhaiten pyöreänmuotoisessa 3D-objektissa, spheremap-projisoinnilla. Lopputuloksena syntyi toimiva tuotantoprosessi ympäristön esittämiseen 3D-virtuaalitiloissa. Valokuvausratkaisussa olisi kuitenkin vielä parantamisen varaa helppoudessa ja tehokkuudessa, sillä radio-ohjattavan kopterin ohjaaminen ei ole mutkatonta ilman harjoittelua, ja kova tuuli voi vaikuttaa kopterin käyttäytymiseen. Valokuvien yhdistäminen voisi olla myös automaattisempaa. Tulevaisuudessa ratkaisu voisi olla esimerkiksi juuri Panono-panoraamakuvauspallo tai 360°-panoraamakuvaobjektiivi, jotka voisivat osaltaan parantaa prosessia.

Lähteet

360 Virtual Tour Cameras. 2014. Verkkodokumentti. Vpix. <http://vpix360.com/vpix-html5-virtual-tour-showcase.php> Luettu 12.4.2014.

3D formats. 2013. Verkkodokumentti. Unity3D Documentation. <http://docs.unity3d.com/Documentation/Manual/3D-formats.html> Luettu 10.4.2014.

3ds Max 3D-mallinnus- ja animointiohjelmisto. 2014. Verkkodokumentti. Autodesk. <http://www.autodesk.fi/products/autodesk-3ds-max/features/all/gallery-view> Luettu 10.4.2014.

About Blender. 2014. Verkkodokumentti. Blender Foundation. <http://www.blender.org/about/> Luettu 10.4.2014.

Ahearn, Luke. 2006. 3D game textures. Burlington: Focal Press.

Autodesk Store. 2014. Verkkodoumentti. Autodesk. <http://store.autodesk.eu/store/adsk/en_IE/html/pbPage.All-Product-Listing_en_IE> Luettu 10.4.2014.

Brown, Nathan. 2010. Verkkodokumentti. Tennis For Two Restored 50 Years On. <http://www.edge-online.com/news/tennis-two-restored-50-years/> Luettu 7.4.2014.

Cad-symbolit. 2013. Verkkodokumentti. Martela. <http://www.martela.fi/cad-symbolit> Luettu 10.4.2014.

100 Most Popular Engines Today. 2014. Verkkodokumentti. DBolical Pty Ltd. <http://www.indiedb.com/engines/top> Luettu 7.4.2014.

Enger, Michael. 2013. Verkkodokumentti. Game Engines: How do they work?. <http://www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-they-work/101529/ > Luettu 7.4.2014.

F360 Explorer. 2014. Verkkodokumentti. Freedom360. <http://freedom360.us/shop/f360-explorer/> Luettu 12.4.2014.

Galuzin, Alex. 2013. Verkkodokumentti. Software for Game Environment Artist. <http://www.worldofleveldesign.com/categories/game_environments_design/software-for-game-environment-artist.php> Luettu 10.4.2014.

Get Skanect . 2014. Verkkodokumentti. ManCTL. <http://skanect.occipital.com/download/> Luettu 12.4.2014.

Harper, Jeffrey M. 2012. Mastering Autodesk 3ds Max 2013. Indianapolis: John Wiley & Sons Inc.

Hero3+ black edition tehnical specs. 2014. Verkkodokumentti. GoPro Inc.
<http://www.dji.com/product/phantom-2/feature> Luettu 12.4.2014.

Kinect for Windows. 2013. Verkkodokumentti. Microsoft. <http://www.microsoft.com/en-us/kinectforwindowsdev/Downloads.aspx> Luettu 10.4.2014.

Kinect for Windows features. 2014. Verkkodokumentti. Microsoft. <http://www.microsoft.com/en-us/kinectforwindows/discover/features.aspx> Luettu 10.4.2014.

Lehtovirta, Pekka & Nuutinen, Kari. 2000. 3D-sisällöntuotannon peruskirja. Jyväskylä:
Docendo

Martela Fly Me -nojatuoli. 2013. Verkkodokumentti. Martela. <http://www.martela.fi/julkitilakalusteet/nojatuolit-ja-sohvat/fly-me-nojatuoli> Luettu 10.4.2014.

Matheson, Rob. 2014. 3-D scanning, with your smartphone. Verkkodokumentti. MIT News. <http://newsoffice.mit.edu/2014/3-d-scanning-with-your-smartphone-0131> Luettu 11.4.2014.

Milian, Mark & Chan, Marcus. 2012. 'Pong' Turns 40, But It's Not the Oldest Video Game. Verkkodokumentti. <http://www.bloomberg.com/slideshow/2012-11-16/-pong-turns-40-but-it-s-not-the-oldest-video-game.html#slide2> Luettu 7.4.2014.

Panoramic ball camera, product features. 2013. Verkkodokumentti. Panono. <http://www.panono.com/features/> Luettu 11.4.2014.

Phantom 2 features. 2014. Verkkodokumentti. Dji. <http://www.dji.com/product/phantom-2/feature> Luettu 12.4.2014.

PhotoModeler Scanner. 2014. Verkkodokumentti. Eos Systems Inc. <http://www.photomodeler.com/products/scanner/default.html> Luettu 10.4.2014.

Puhakka, Antti. 2008. 3D-grafiikka. Helsinki: Talentum

Realviz Releases ImageModeler 3.5 for Windows. 2003. Verkkodokumentti. The CGSociety. <http://www.cgsociety.org/index.php/CGSFeatures/CGSFeatureSpecial/realviz_releases_imagemodeler_3.5_for_windows> Luettu 10.4.2014.

St. John, Alex. 2013. The Evolution of Direct3D. Verkkodokumentti. <http://www.alexstjohn.com/WP/2013/07/22/the-evolution-of-direct3d/> Luettu 7.4.2014.

The History of DirectX. 2010. Verkkodokumentti. CodingUnit. <http://www.codingunit.com/the-history-of-directx> Luettu 9.4.2014.

The History of OpenGL. 2010. Verkkodokumentti. CodingUnit. <http://www.codingunit.com/the-history-of-opengl> Luettu 8.4.2014.

Unity Technologies. 2014. Verkkodokumentti. Unity3D store. <https://store.unity3d.com/> Luettu 7.4.2014.

Ward, Jeff. 2008. What is a Game Engine? Verkkodokumentti. <http://www.gamecareerguide.com/features/529/what_is_a_game.php/ > Luettu 30.3.2014.

Wave hello to Microsoft's Kinect. 2010. Verkkodokumentti. Guardian News and Media. <http://www.theguardian.com/technology/2010/jun/15/kinect-xbox-microsoft-e3> Luettu 11.4.2014.

3D realtime model requirements

Here is a list of requirements for the exported models.

LOD(level of detail) models:

Very small and unimportant models should not contain level of detail versions and use only low poly count. Charachters and large buildings should contain atleast low to medium polycount LOD models. Very large object and important objects should contain all of the LOD models.

Examples: character contains three version of itself with different polycount. Low, medium and large. Small object contains only one model, which is low detailed. A very large house or building contains all of the LOD models

LOD triangle counts:

- low 0 - 299 triangles(899 vertices) <- requirement for dynamic batching
- medium 299 – 750 triangles
- high poly count max 750 – 1500 triangles for large objects and charachters
- 1500 – 4500 triangles for very very large objects

Check list for objects to be exported:

- Single object has only one material
- Exported models should not be triangulated, because somebody might need to modify it.
- Non important static meshes in close proximity in to one uv atlas
- reuse non important models repetitive texture areas
- Pivots centered and objects put to stand on top and center of the origo
- Two uv sets. One for diffuse and one for lightmaps. The second uv-map can be generated in unity import settings.
- Vertices merged together
- Texture format tiff(layered), non layered png or tga

• Models exported as obj or fbx depending, if they have animations or not
• Texture size max 4096 for very large objects
• Texture size max 2048 for medium to low objects
• exported model should also contain texture converted to 1024 resolution
• Textures bigger than 1024 to a folder outside unity project. This way project conversion times are lower
• Textures power of two. Example 32, 64, 128, 256, 512, 1024, 2048, 4096 etc...

Dynamic And Static Batching

These are techniques to increase real-time rendering performance.

Dynamic Batching

Unity can automatically batch moving objects into the same draw call if they share the same material and fulfill other criteria. Dynamic batching is done automatically and does not require any additional effort on your side. However models need to meet the following requiments. Batching dynamic objects has certain overhead per vertex, so batching is applied only to meshes containing less than 900 vertex attributes in total. If your shader is using Vertex Position, Normal and single UV, then you can batch up to 300 vertices. whereas if your shader is using Vertex Position, Normal, UV0, UV1 and tangent, then only 180 verts. Please note: attribute count limit might be changed in future Generally, objects should be using the same transform scale. The exception is non-uniform scaled objects; if several objects all have different non-uniform scale then they can still be batched. Using different material instances - even if they are essentially the same - will make objects not batched together. Objects with lightmaps have additional renderer parameter: lightmap index and offset/ scale into the lightmap. So generally dynamic lightmapped objects should point to exactly the same lightmap location to be batched.Multi-pass shaders will break batching. Almost all unity shaders supports several lights in forward rendering, effectively doing additional pass for them. The draw calls for "additional

per-pixel lights" will not be batched. Objects that receive real-time shadows will not be batched.

Static Batching

Static batching, on the other hand, allows the engine to reduce draw calls for geometry of any size (provided it does not move and shares the same material). Static batching is significantly more efficient than dynamic batching. You should choose static batching as it will require less CPU power. In order to take advantage of static batching, you need explicitly specify that certain objects are static and will not move, rotate or scale in the game. To do so, you can mark objects as static using the Static check box in the Inspector: Using static batching will require additional memory for storing the combined geometry. If several objects shared the same geometry before static batching, then a copy of geometry will be created for each object, either in the Editor or at runtime. This might not always be a good idea - sometimes you will have to sacrifice rendering performance by avoiding static batching for some objects to keep a smaller memory footprint. For example, marking trees as static in a dense forest level can have serious memory impact.Static batching is only available in Unity Pro for each platform.

Sources:

http://docs.unity3d.com/Documentation/Manual/

http://docs.unity3d.com/Documentation/Manual/DrawCallBatching.html

Author: Mikael Korpinen 16.7.2013