

3D-mallin sekä valaistuksen toteutus asuntoesittelysovellukseen

Markus Palomäki

Opinnäytetyö

Toukokuu 2017

Tekniikan ja liikenteen ala

Insinööri (AMK), mediatekniikan koulutusohjelma

Tekijä(t) Palomäki, Markus Veli Johannes	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Toukokuu 2017
	Sivumäärä 37	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi 3D-mallin sekä valaistuksen toteutus asuntoesittelysovellukseen		
Tutkinto-ohjelma Mediatekniikka		
Työn ohjaaja(t) Kari Niemi		
Toimeksiantaja(t) Zaibatsu Interactive Oy		
<p>Tiivistelmä</p> <p>Opinnäytetyön tavoitteena oli tutkia sekä luoda ohjeistus 3D-mallin ja valaistuksen toteuttamisesta Unity3D-pelimootorilla ajettavaan asuntoesittelysovellukseen. Asuntoesittelysovelluksessa pääsee tutustumaan suunnittelemaan uudisrakennuskohteeseen, jota voitiin ajaa verkkoselaimella.</p> <p>Opinnäytetyössä 3D-malliksi valittiin divaanisänky. Sängyn prosessi aloitettiin objektin suunnittelusta jatkuen aina low poly -matalaresoluutioisen mallin toteutukseen sekä mallin viemiseen Unity3D-kehitystyökaluun. Sängyn tyynyistä ja peitosta luotiin korkearesoluutioiset (high poly) mallit, jotka toteutettiin käyttämällä Blender-sovelluksen fysiikkamootoria. Korkearesoluution malleista luotiin edelleen matalaresoluutioiset (low poly) mallit.</p> <p>Asuntoesittelysovelluksessa valaistus oli jo valmiiksi toteutettuna, mutta asiakas halusi kehittää sitä valoisammaksi. Tämä saatiin toteutettua käyttämällä eri Unity3D-sovelluksen valaistusmenetelmiä.</p> <p>Opinnäytetyön tuloksena saatiin toteutettua sängystä 3D-malli aikuisten makuuhuoneeseen. Sovellus sai valoisamman ja realistisemman valaistuksen. Lisäksi opinnäytetyön kirjallista osiota voidaan käyttää uusien mallien toteuttamiseen, mallin viemiseen Unity3D-pelimootoriin sekä valaistuksen asettamiseen.</p>		
Avainsanat (asiasanat) 3D-mallinnus, valaistus, low poly, high poly, Blender, Unity3D		
Muut tiedot		

Author(s) Palomäki, Markus Veli Johannes	Type of publication Bachelor's thesis	Date May 2017
	Number of pages 37	Language of publication: Finnish
	Permission for web publication: x	
Title of publication Implementing 3D model and lighting in a residential demo application		
Degree programme Media Engineering		
Supervisor(s) Kari Niemi		
Assigned by Zaibatsu Interactive Oy		
Abstract <p>The aim of the bachelor's thesis was to study and create guidelines for the implementation of the 3D model and lighting in a residential demo application. The application runs on Unity3D game engine. In the residential demo application, a player can get acquainted with a new building. The application could be run on a web browser.</p> <p>In the thesis, a divan bed was chosen as the 3D model. The process of creating the bed was started from the design of the object, continuing to implement a low poly model and to import the model into the Unity3D game engine. The bed's pillows and blanket were created in high poly model which were made by using the physics engine of the Blender application. The low poly models were created by using high poly models.</p> <p>In the residential demo application, the lighting was already completed, however, the client wanted to develop it to be brighter. This was accomplished using Unity3D's different lighting methods.</p> <p>As a result of the thesis, the model of the divan bed was finished and placed into the master bedroom. The application got a brighter and more realistic lighting. In addition, the written part of the thesis can be used to implement new models, to import models into Unity3D game engine and to set the lighting.</p>		
Keywords/tags (subjects) 3D modeling, lighting, low poly, high poly, Blender, Unity3D		
Miscellaneous		

Sisältö

1	Työn lähtökohdat	4
1.1	Tausta ja toimeksiantaja.....	4
1.2	Tavoitteet	4
2	3D-grafiikka	5
2.1	3D-ohjelmat.....	5
2.1.1	Autodesk.....	6
2.1.2	Blender.....	7
2.2	Mallintaminen	8
2.2.1	Topologia	8
2.2.2	High poly ja low poly.....	9
2.2.3	Mallin luominen ja muokkaaminen.....	10
2.2.4	Modifiers.....	11
2.2.5	Sculpting	12
2.2.6	Physics.....	12
2.3	Materiaalit	13
2.3.1	Materiaalin ominaisuudet	13
2.3.2	Normal map	14
2.3.3	UV-kartoitus.....	15
2.4	Renderointi.....	16
2.4.1	Yleistä.....	16
2.4.2	Renderöntimoottorit	17
2.5	Tiedoston vienti.....	17
2.5.1	Yleistä.....	17
2.5.2	FBX-tiedosto	17
3	Unity3D.....	18
3.1	Objektin lisääminen.....	19

	2
3.2	Objektin asetukset..... 19
3.2.1	Yleistä..... 19
3.2.2	Materiaalit 20
3.3	Valaistus 21
4	CASE: Asuntoesittelysovellus 22
4.1	Sängyn toteuttaminen..... 22
4.1.1	Lähtökohdat ja suunnittelu 22
4.1.2	Mallintaminen 22
4.1.3	Mallin vieminen tiedostoon..... 29
4.1.4	Objektin lisääminen asuntoesittelysovellukseen 30
4.1.5	Materiaalin asettaminen 30
4.2	Valaistus 31
5	Tulokset ja pohdinta 33
	Lähteet 36

Kuviot

Kuvio 1.	Koordinaattiakselit x, y ja z..... 5
Kuvio 2.	Ruudunkaappaus Autodeskin 3ds Max 2016 -sovelluksesta 6
Kuvio 3.	Ruudunkaappaus Blender-sovelluksesta..... 7
Kuvio 4.	Hexagon-pallo pisteillä, reunoilla ja pinnoilla 8
Kuvio 5.	Pallon muotoiset low-poly sekä high-poly -mallit..... 9
Kuvio 6.	Työkalujen Extrude ja Fill tuloksia 10
Kuvio 7.	Kahden muotoilun järjestyksen muuttaminen ja sen vaikutukset 11
Kuvio 8.	Sculpting Mode -näkyvän pensseleitä sekä niiden vaikutukset (Tools n.d.) 12
Kuvio 9.	Esimerkkejä erilaisista pinnoista tosielämässä 13
Kuvio 10.	Diffuse-, glossy- ja specular-heijastukset 14
Kuvio 11.	Esimerkki normal map -kuvasta 14
Kuvio 12.	Kuinka Blender tulkitsee 3D-kuution unwrappauksen jälkeen 15

Kuvio 13. Ruudunkaappaus Unity-kehitystyökalun käyttöliittymästä	18
Kuvio 14. Inspector-ikkuna	20
Kuvio 15. Unity-sovelluksen kolme valotyyppiä.....	21
Kuvio 16. Näyttökaappaus Blender-sovelluksen Units-näkymästä.....	23
Kuvio 17. Näyttökaappaus Blender-sovelluksen Physics-ikkunan Cloth-näkymästä...	24
Kuvio 18. Tyyny-mallit sekä pintojen lukumäärät.....	26
Kuvio 19. Decimate-työkalun Collapse (vasemmalla) ja Un-Subdivide (oikealla).....	28
Kuvio 20. Valmiiksi saatu sänky-malli Blender-sovelluksessa renderöitynä.	28
Kuvio 21. Mallien vientiasetukset.	29
Kuvio 22. Materiaalin ominaisuuksia Unityssa.....	30
Kuvio 23. Asuntoesittelysovelluksen keittiö	32
Kuvio 24. Toinen huoneisto valaistuksen jälkeen	33
Kuvio 25. Mallinnettu sänky sekä säädetty valaistus asuntoesittelysovelluksessa	34

1 Työn lähtökohdat

1.1 Tausta ja toimeksiantaja

Opinnäytetyön toimeksiantajana toimi Zaibatsu Interactive Oy (myöhemmin Zaibatsu). Jyväskylässä keväällä vuonna 2014 perustettu yritys keskittyy pelikehityksen lisäksi tarjoamaan yrityksille pelillistämistä sekä sovelluskehitystä. Yritys työllistää 9 henkilöä.

Zaibatsu julkaisi Suomessa lokakuussa 2015 yhdessä Yleisradion kanssa Elder Goo seikkailupulmapelin. Yhden laitteen monipelissä pelaaja tai pelaajat ratkovat erilaisia tasoja neljällä pelattavalla mölli-hahmolla, joilla jokaisella on oma erikoiskykynsä. Peli tullaan julkaisemaan maailmanlaajuisesti 2017 vuoden loppupuolella (Zaibatsu Interactive Inc.).

Opinnäytetyö tehtiin Zaibatsun asiakasprojektin yhteydessä. Uudisrakennuskohteet tullaan rakentamaan kesällä 2017 alkaen. Zaibatsu toteuttaa asiakasprojektina virtuaalisen asuntoesittelysovelluksen, jonka avulla asunnoista kiinnostuneet voivat tutustua asuntoihin verkkosivun kautta. Sovellus on toteutettu Unity3D-pelimootorilla.

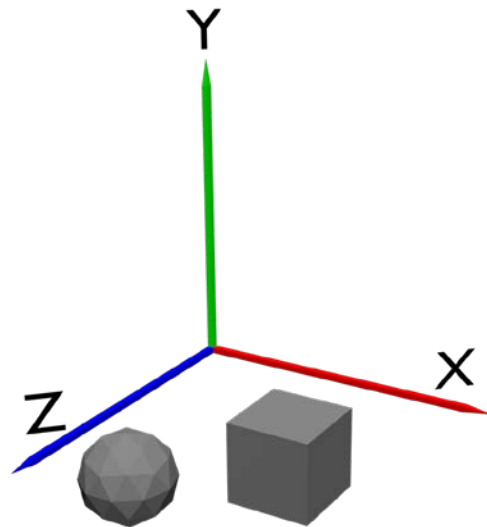
1.2 Tavoitteet

Opinnäytetyön tavoitteena oli tutkia 3D-mallintamista ja valaistuksen asettamista Unity3D-pelimootorissa. Haasteina olivat muun muassa sovelluksen ajaminen verkkosivulla sekä toimivuus mahdollisimman monella laitteella.

Opinnäytetyössä käydään eri vaiheet ohjeistusmuodossa. Opinnäytetyössä tutkitaan 3D-mallintamista käyden mallintamisen prosessi alkaen suunnittelusta toteutukseen, vähäpintaisen version tuottamiseen, teksturointiin ja lopulta valmiin objektin lisääminen sovellukseen. Lisäksi opinnäytetyössä tutkitaan valaistuksen asettamisesta Unity3D-sovellukseen ja sen mukauttamisesta.

2 3D-grafiikka

3D-grafiikka luodaan käyttämällä kolmea koordinaattiakselia. 2D-koordinaatisto hyödyntää koordinaattiakseleita x ja y , jota hyödynnetään muun muassa verkkosivujen toteuttamisessa. X -koordinaattiakseli on horisontaalinen ja y -koordinaattiakseli vertikaalinen. 3D-grafiikassa lisätään uusi koordinaattiakseli z , joka kuvaa syvyyttä (ks. kuvio 1). Syvyydellä tarkoitetaan sitä, kuinka kaukana katsojasta 3D-malli piirretään esimerkiksi 3D-ohjelmassa. (Parisi 2014, 9.)



Kuvio 1. Koordinaattiakselit x , y ja z

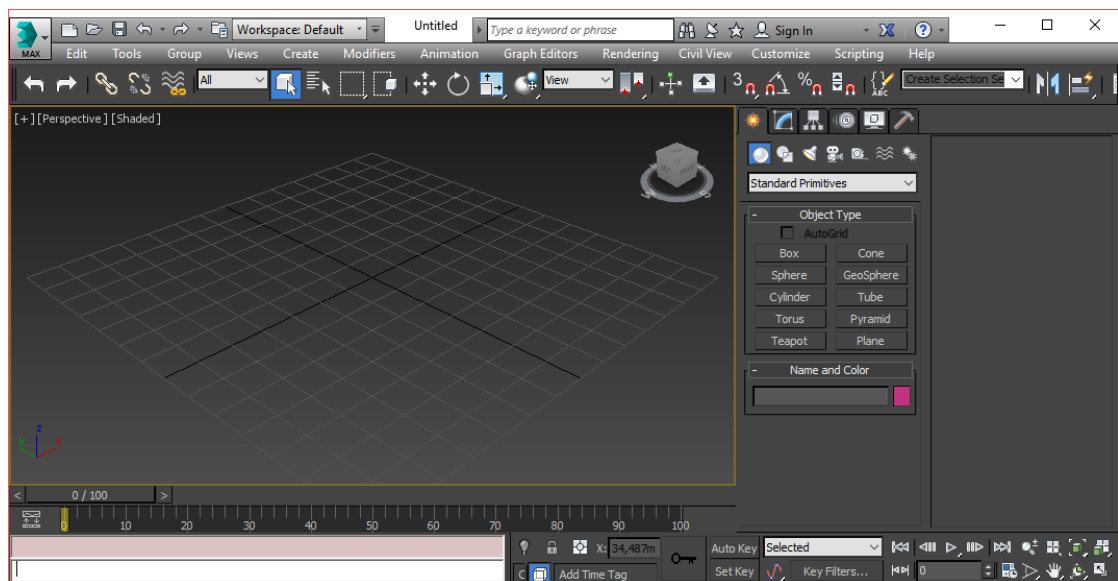
2.1 3D-ohjelmat

3D-mallintamisessa vaaditaan esimerkiksi tietokoneella ajettavia sovelluksia, jotka ovat räätälöityjä juuri 3D-grafiikan luomiseen. Sovelluksen avulla saadaan luotua geometrinen malli, joka koostuu geometrisista pinnoista ja kappaleista. Sovelluksien avulla saadaan 3D-malleille asetettua muun muassa valaistuksia ja tekstuureja. (Puhakka 2008, 428-429.)

3D-ohjelmia löytyy sekä maksullisina että ilmaisversioina. Lisäksi osa 3D-mallinnusohjelmista keskittyy tiettyyn mallintamisen osa-alueeseen kuten animaation luomiseen.

2.1.1 Autodesk

Autodesk tarjoaa kattavasti sovelluksia arkkitehti- ja rakennesuunnitteluun, tuotesuunnitteluun sekä median luomiseen (ks. kuvio 2). Jälkimmäisenä mainittuun median luomiseen käytettäviin työkaluihin Autodesk luettelee animaattorien, mallintajien ja tehostetaiteilijoiden käyttämiä työkaluja kuten Maya ja 3ds Max. Sovelluksien monipuolisuudesta kertoo se, että Maya-sovellusta ollaan käytetty muun muassa Disney-elokuvan Viidakkokirjan tekemiseen. (Kattavat luovat työkalut viihdealan suunnittelua varten n.d.)



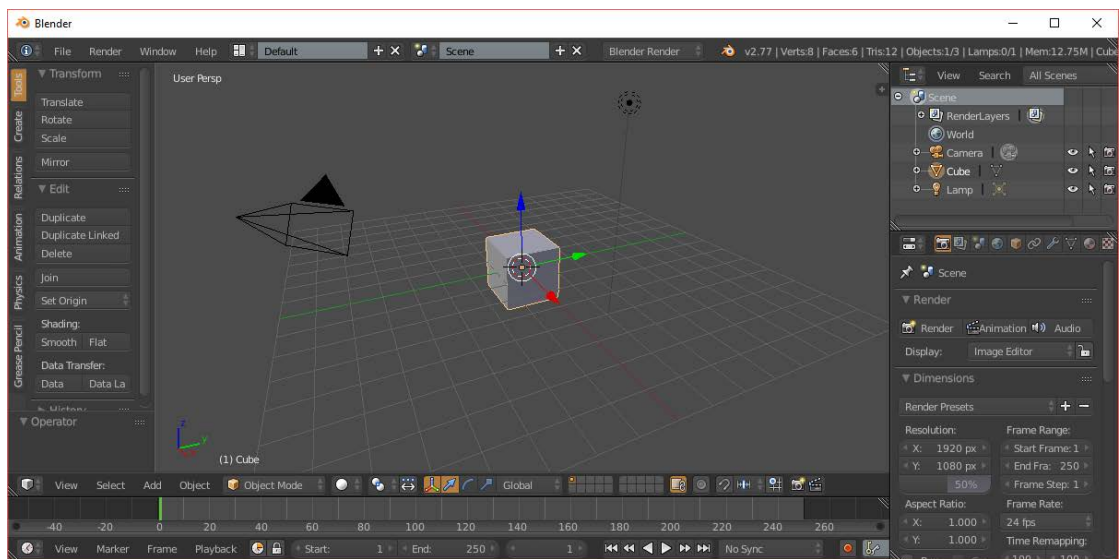
Kuvio 2. Ruudunkaappaus Autodeskin 3ds Max 2016 -sovelluksesta

Jyväskylän Ammattikorkeakoululta on mahdollisuus saada 3ds Max -sovellus käytettäväksi oppimistarkoitukseen Educational-lisenssillä. Educational-lisenssi on tarkoitettu oppimis-, opettamis-, harjoittelu-, tutkimus- ja kehittämikäyttöön. Kyseisellä lisenssillä ei saa tehdä töitä kaupalliseen, ammattilais- tai voittoa tavoiteltaviin tarkoituksiin. (License and Services Agreement n.d.)

2.1.2 Blender

Blender on ilmainen ja avoimen lähdekoodin 3D-mallintamisohjelma, jota käyttämällä käyttäjä voi mallintaa, riggata, animoida, simuloida, sekä renderöidä (ks. kuvio 3) (About n.d.). Blender-sovellusta jaetaan GNU General Public License -lisenssin alla, joka antaa sovelluksen käyttäjälle oikeuden:

- käyttää Blender-sovellusta mihin tarkoitukseen tahansa
- jakaa sovellusta
- tutkia kuinka Blender-sovellus toimii ja muokata sitä
- jakaa muokattua versiota Blender-sovelluksesta.



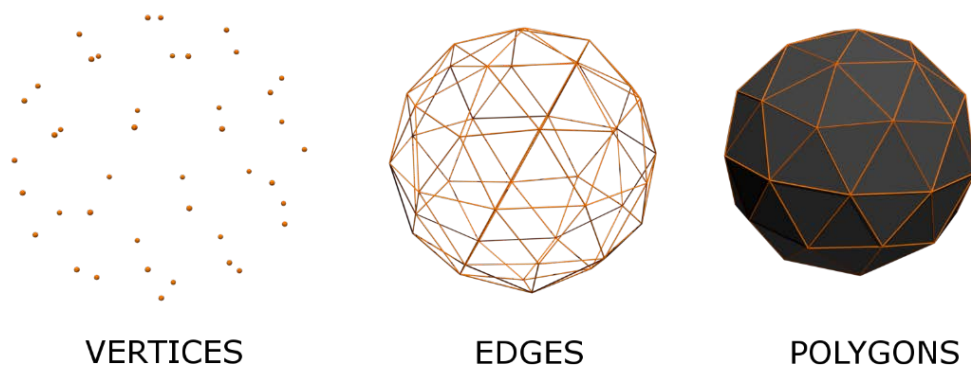
Kuvio 3. Ruudunkaappaus Blender-sovelluksesta

Blender-sovelluksen avoimuuden ansiosta kaikki sovelluksella tehdyt työt ovat tekijän omaisuutta. Kaikki tiedostot kuten sovelluksella renderöidyt kuvat, videot ja muut Blender-sovelluksen kirjoittamat tiedostot ovat vapaasti tekijän käytettävissä. Tämän ansiosta sovellusta voidaan käyttää tutkimus-, koulutus- tai kaupallisessa käytössä. (License n.d.)

Syy Blender-sovelluksen käyttöön on sen ilmaisuus ja avoin lähdekoodi. Zaibatsu pysyy Blenderin avulla jatkokehittämään, sillä luotuja malleja, kun taas Autodesk-sovelluksen Educational-lisenssi ei olisi mahdollistanut tätä.

2.2 Mallintaminen

3D-malli luodaan käyttämällä pisteitä 3D-avaruudessa. Pisteiden (englanniksi vertex points tai vertices) välille luodaan viivoja, joista muodostuu reunat. Reunat (englanniksi edges) luovat objektille verkkomaisen muodon. Lisäämällä tasuja (englanniksi faces tai polygons) reunoilla suljetun alueen sisälle saadaan objektille kiinteä pinta (ks. kuvio 4). Pinnat voidaan värittää, jotta saadaan realistiset tekstuurit objektille. (Flavell 2010, 38.)



Kuvio 4. Hexagon-pallo pisteillä, reunoilla ja pinnoilla

2.2.1 Topologia

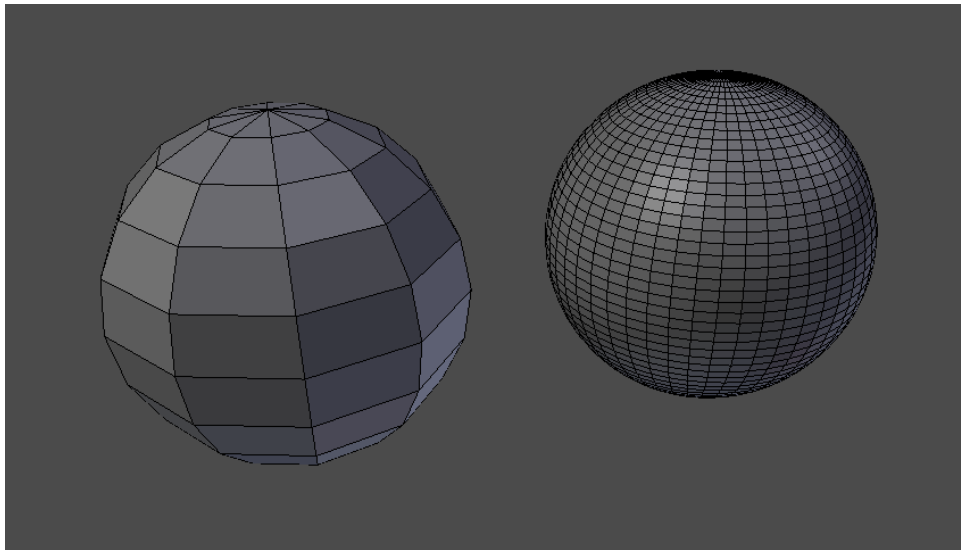
Objekteja mallintaessa on hyvä miettiä mallien topologiaa. Topologialla viitataan 3D-mallin geometrisen muodon ominaisuuksiin kuten pintojen reunojen asetteluun, edgeloopsiin. Etenkin animaatiossa on tärkeää pohtia pintojen reunojen asettelua, jotta animaation aikana ei tapahtuisi vääristymiä pinnoissa. (Flavell 2010, 47; Slick 2016.)

Topologia voi olla toteutettu hyvin tai huonosti riippuen siitä, kuinka hyvin pintojen reunaviivat seuraavat mallin pinnan muotoja. Lisäämällä pintojen lukumäärää tai pehmentämällä pinnan muotoa käyttämällä Modifiers -työkalua, objektin pinnalle voi ilmestyä ei-haluttuja vääristymiä. Myös pintojen muodot (kolmion-, neliönmalliset tai monikulmioiset) vaikuttavat. Etenkin animaatiossa kappaleen taittuminen voi epäonnistua valitsemalla mallille vääränlaiset pinnat kuten kolmionmallisia pintoja. (Flavell, 2010.)

2.2.2 High poly ja low poly

High poly ja low poly ovat termejä, joiden avulla kerrotaan objektin tarkkuudesta. High poly -malleilla on paljon pintoja, jonka avulla malleista saadaan yksityiskohtaisempia. Tällaisia malleja käytetään lähikuvissa. High poly -malleja käytettäessä yksityiskohtiin panostetaan eikä suorituskykyyn, sillä yksityiskohtaisempi objekti vaatii enemmän resursseja sitä pyörittävältä järjestelmältä (Derakhshani & Munn 2008, luku 6).

Objektia, jolla on matala pintalukumäärä, kutsutaan low poly -malliksi. Verrattuna high poly -malliin low poly -mallilla on vähemmän pintoja, jolloin pintojen vähyyden takia malli ei ole yhtä yksityiskohtainen kuin high poly -malli (ks. kuvio 5). Low poly -mallit rasittavat järjestelmää – kuten tietokonetta – vähemmän, minkä takia kyseiset mallit ovat yleisesti käytettyjä videopeleissä. (Derakhshani & Munn 2008, luku 6.)



Kuvio 5. Pallon muotoiset low-poly sekä high-poly -mallit

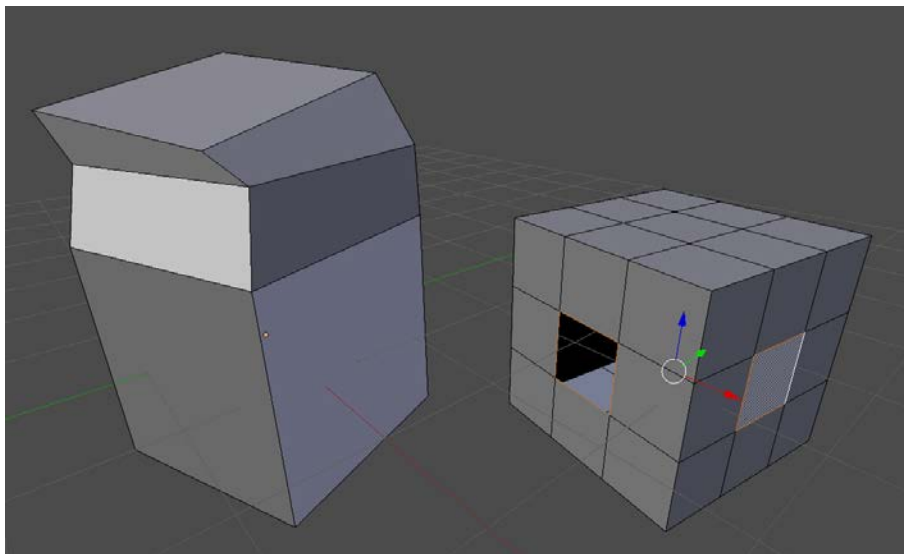
2.2.3 Mallin luominen ja muokkaaminen

3D-mallinnussovellukset tarjoavat valmiiksi mallikirjaston, joka sisältää yksinkertaisia malleja. Blender-sovellus sisältää perusmalleja, kuten kuutioita, putkia ja palloja, jotka ovat käytännöllisiä mallintamista aloittaessa (Flavell 2010, 23).

Objektia pystyy luomisen jälkeen muokkaamaan muun muassa liikuttamalla, kääntämällä ja skaalaamalla. Liikuttamalla objekti liikkuu 3D-avaruudessa, objekti voidaan kääntää haluttuun asentoon ja skaalauksella voidaan suurentaa tai pienentää objektia. (Flavell 2010, 25.)

Blender-sovelluksen Edit -näkyvä saadaan malleja muokattua monipuolisemmin. Näkymän ollessa aktiivisena muokkaus kohdistuu objektiin, joka oli aktiivisena työkalun käyttöönoton aikana. Tämän aikana voidaan siirtää, skaalata ja kääntää yhtä tai useaa vertex-pistettä, reunaa tai pintaa. (Flavell 2010, 39-40). Lisäksi valittuja elementtejä voidaan poistaa tai yhdistää käyttämällä Delete- ja Merge-työkaluja (Flavell 2010, 45).

Edit -näkyvässä valittuja elementtejä saadaan työnnettyä ulospäin käyttämällä Extrude-työkalua. Tällöin valittu elementti ei liiku, vaan valitun elementin jatkoksi luodaan uutta elementtiä. Fill-työkalulla saadaan luotua tasoja valittujen vertex-pisteiden tai reunojen sisälle (ks. kuvio 6). (Flavell 2010, 42-43.)



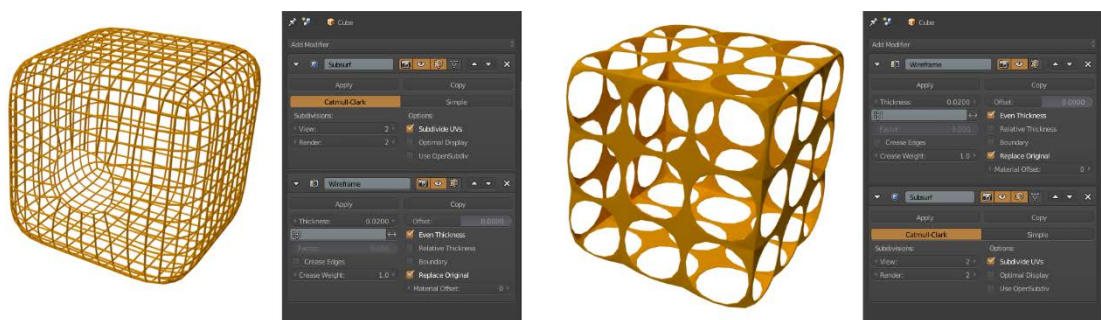
Kuvio 6. Työkalujen Extrude ja Fill tuloksia

2.2.4 Modifiers

Mallintaminen on lopulta yksinkertaista, mutta joskus se voi olla raskasta. Tietyn muodon mallintaminen tai toteuttaminen voi viedä paljon aikaa. Tätä varten Blender-sovelluksesta löytyy työkalu Modifiers (suomeksi muotoilu), joiden avulla mallintamista saadaan helpotettua ja nopeutettua. Samalla muotoilutyökalut ovat helppo tapa luoda ja testaila tuloksia, sillä vasta muotoilun hyväksyminen asettaa muokkaukset malliin (Flavell 2010, 52).

Käyttäessä Blender-sovelluksen muotoilutyökaluja on hyvä miettiä eri työkalujen käyttöä objektin mallintamiseen. Blender-sovelluksessa muotoilutyökalut toimivat filtterin tavoin; monen muotoilun järjesteleminen muuttaa lopputulosta. Tämä on hyvä muistaa muotoiluja hyväksyttäessä, sillä väärässä järjestyksessä muotoilujen hyväksyminen vaikuttaa lopputulokseen. (Flavell 2010, 52.)

Esimerkkinä kuviossa 7 nähdään kahden muotoilun lopputulokset eri järjestyksessä. Vasemmalla Subsurf-työkalu kaksinkertaistaa pintojen lukumäärän, mikä samalla pyöristää objektia. Tämän jälkeen pintojen reunoista luodaan verkkomainen rakenne käyttämällä Wireframe-työkalua. Kuvassa oikealla samat työkalut, mutta eri järjestyksessä. Wireframe-työkalu luo low-poly mallille verkkomaisen rakenteen, jonka jälkeen Subsurf-työkalulla lisää pintoja samalla pyöristäen verkkomaisen rakenteen kulmia. (ks. kuvio 7.)

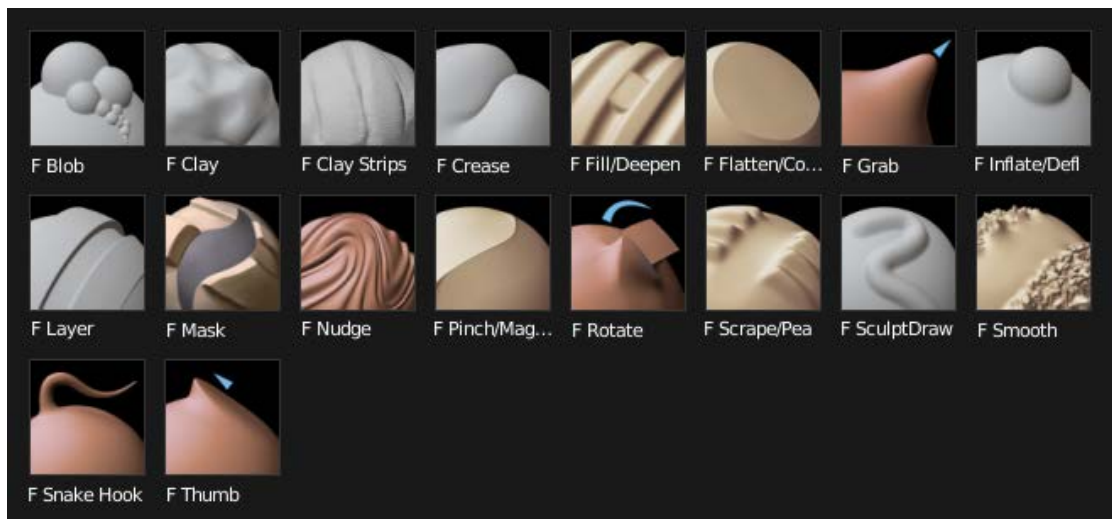


Kuvio 7. Kahden muotoilun järjestyksen muuttaminen ja sen vaikutukset

2.2.5 Sculpting

Blender-sovelluksessa voidaan mallintaa objekteja savimuovailun tavoin. Sculpt Mode -näkömön avulla voidaan luoda objekteja käyttämällä eri työkaluja. Sculpt -näkömön on kuin Edit Mode -näkömön, mutta sen työkalu on erilainen. Sculpt Mode -näkömön yksittäisten elementtien (vertex, edges, faces) käsittelysijain malli luodaan alkeellisesta mallista, jota muokataan käyttämällä pensseleitä. Pensseleitä löytyy erilaisia, joista jokainen käyttäytyy eri tavalla. (Flavell 2010, 55; Introduction n.d.)

Pensseleitä on monenlaisia. Esimerkiksi Smooth-pensselillä voidaan poistaa epätasaisuutta pensselillä valitulta alueelta tasoittamalla pisteiden (vertices) sijaintia. Kuviossa 8 nähdään muita Sculpting Mode -näkömön pensseleitä ja niiden vaikutuksia.



Kuvio 8. Sculpting Mode -näkömön pensseleitä sekä niiden vaikutukset (Tools n.d.)

2.2.6 Physics

Blender-sovelluksen sisäänrakennetulla fysiikkamoottorilla voidaan simuloida erilaisia oikean maailman fyysisiä ilmiöitä. Blender-sovelluksella voidaan toteuttaa muun muassa nesteitä, savuja, voimia, hiuksia ja kankaita. (Introduction n.d.)

Blender-sovelluksen fysiikkamoottori hyödyntää painovoimaa simuloimassa fyysisiä ilmiöitä. Gravitaatioksi on asetettu arvoksi 9.810 joka on maapallolla vallitseva putoamiskiihtyvyyden arvo vaikuttaa korkeusakseliin Z. Muuttamalla arvoa voidaan simuloida esimerkiksi Kuun painovoimaa asettamalla gravitaatioksi -1.622. (Gravity n.d.)

Ennen renderöintiä fysiikkasimulaatiot olisi hyvä bakettaa, mikäli joutuu 3D-ympäristön renderöimään uudelleen. Fysiikkasimulaatiossa baketuksella (englanniksi baking) tarkoitetaan fysiikoiden laskutuloksien tallentamista välimuistiin, jolloin fysiikoita ei tarvitse laskea aina renderöinnin aikana uudelleen. Lisäksi bakettuna fysiikoilla luodut animaatiot pysyvät samoina ja mahdollisia virheitä ilmaantuu vähemmän. (Baking Physics Simulations n.d.)

Cloth Dynamics -fysiikan avulla voidaan simuloida muun muassa eri kankaita, vaatteita, lippuja ja bannereita. Kangas saadaan reagoimaan muihin liikkuviin objekteihin, fysikaalisiin ilmiöihin kuten tuuleen ja muihin voimiin. (Introduction n.d.)

2.3 Materiaalit

Malleilla on muodon lisäksi väritys tai kuvio pinnalla, kuten oikeassa elämässä. Pöydällä voi olla puu-kuviointi, juomapullossa logo tai tyynyliinassa ruutukuviointi (ks. kuvio 9). Väritys tai kuvio voidaan toteuttaa asettamalla erilaisia materiaaleja 3D-mallin pinnalle. Valon osuessa pintaan valo voi heijastua takaisin, kulkea pinnan läpi tai imeytyä pintaan. Objektin pinnalle määritetään väri, heijastavuus, läpinäkyvyys ja muita ominaisuuksia. (Connell 2008, luku 6.)

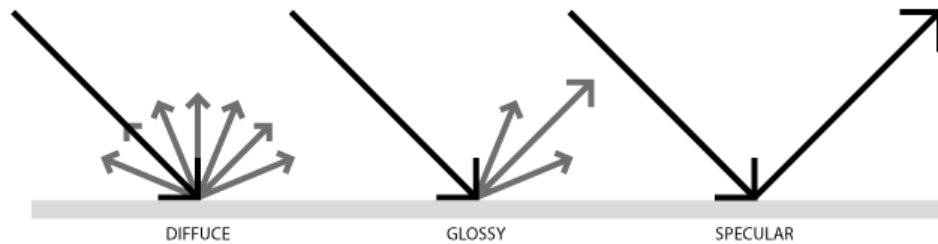


Kuvio 9. Esimerkkejä erilaisista pinnoista tosielämässä

2.3.1 Materiaalin ominaisuudet

Yksinkertaisin materiaali, jonka voi asettaa malliin, on väri. Objektille asetettavat värit voivat muuttua käytettävän valaistuksen kanssa. Lopullinen väri, joka välittyy katsojalle, riippuu valaistuksesta. Esimerkiksi auto valkoisen värisellä materiaalilla voi näyttää siniseltä, jos siihen kohdistettu valaistus on väriltään sininen. Lisäksi materiaalille voidaan määrittää monia ominaisuuksia kuten läpinäkyvyyden, heijastavuuden ja kiiltävyyden (Murdock 2011, luku 15).

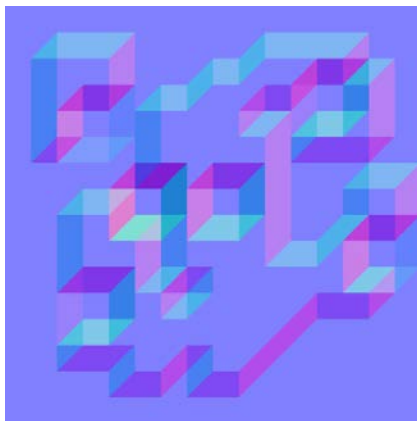
Kolme yleisintä, 3D-sovelluksissa käytettävää valon heijastumistapaa materiaalin pinnasta ovat diffuse, glossy ja specular. Diffuse-heijastus hajottaa valon jokaiseen suuntaan. Glossy-heijastus hajottaa valoa kuten diffuse mutta valon menosuuntaan. Specular-heijastus heijastaa täydellisesti valon tarkkana ilman hajontaa. Näistä diffuse- ja glossy-heijastukset aiheuttavat hajontaa tai heijastuksen sumenemista (ks. kuvio 10). (Brin 2014, 312.)



Kuvio 10. Diffuse-, glossy- ja specular-heijastukset

2.3.2 Normal map

Normal map -kuva lisää malleihin yksityiskohtia tekstuurin avulla, jolloin mallin pinnanmuodot näyttävät yksityiskohtaisemmalta, mitä oikeasti ovat. Tämän avulla vähäpintaiseen malliin saadaan tehtyä esimerkiksi painaumia, halkeamia ja kohoamia muuttamatta mallin muotoa. (Flavell 2010, 114.)



Kuvio 11. Esimerkki normal map -kuvasta

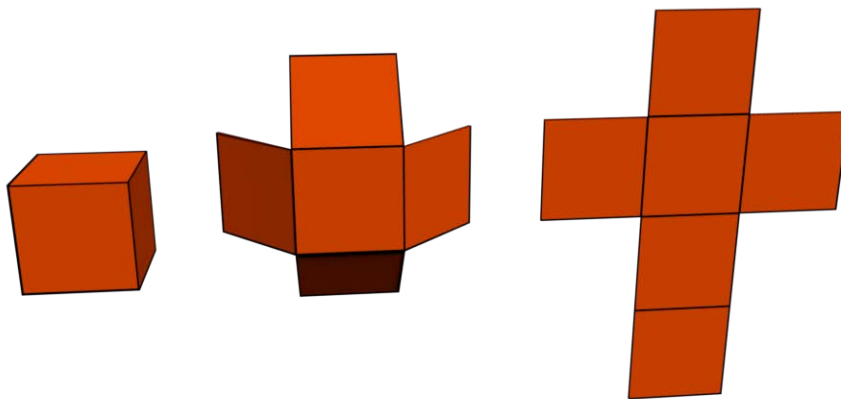
Normal map -kuva toimii käyttämällä punaista, vihreää ja sinistä väriä (ks. kuvio 11). Jokainen väri kertoo sovellukselle, kuinka pinta on vinossa tietyistä kulmista katsotuna. Kolmea väriä käyttämällä normal map -kuvat ovat tarkempia kuin bump map -kuvat, jotka luovat tekstuurin normal map -kuvan tavoin käyttämällä ainoastaan harmaasävyjä. (Flavell 2010, 114.)

2.3.3 UV-kartoitus

UV-kartoitus (UV mapping) on vaihe asettaa tekstuurit objektiin käsin. Tekstuuri haetaan kuvatiedostosta, joka siirretään mallin pinnalle. (Flavell 2010, 97-98.) Tätä varten mallista pitää luoda UV map layout, jotta 3D-malli saadaan siirrettyä kuvatiedostoon tai -tiedostoihin.

Esimerkiksi kuutio saadaan UV-kartoitettua valitsemalla Edit Mode -näkyvässä halutut reunat ja valitsemalla Mark Seam -valinta. Tällöin valittuihin reunoihin tehdään leikkaukset, joiden mukaan objekti saadaan avattua teksturoitavaksi. Tämän jälkeen malli avataan Blender-sovelluksessa käyttämällä Unwrap -työkalua.

Kuviossa 12 kuutio mallinnettuna alkuperäisenä ja vaiheella avattuna. Oikeanpuoleinen malli kuvastaa, miten Blender tulkitsee kuution tekstuurin unwrappauksen jälkeen.



Kuvio 12. Kuinka Blender tulkitsee 3D-kuution unwrappauksen jälkeen

2.4 Renderöinti

2.4.1 Yleistä

Terminä renderöinti (englanniksi rendering) tarkoittaa 3D-sovelluksen laskelmia muuttaessaan 3D-ympäristö viimeistellyksi 2D-kuvaksi. Prosessin aikana 3D-ympäristön alue-, tekstuuri- ja valaistustiedot yhdistetään ja niiden väriarvot tallennetaan kuvaan kukin pikseli kerrallaan. Renderöintiin voidaan lukea kaksi eri tyyppiä, jotka ovat reaaliaikainen renderöinti (real-time rendering) ja esirenderöinti (pre-rendering). (Slick 2017.)

Reaaliaikaista renderöintiä käytetään etenkin videopeleissä sekä interaktiivisessa grafiikassa, joissa 3D-informaatio joudutaan laskemaan nopealla syklillä. Esimerkiksi videopeleissä reaaliaikaista renderöintiä tarvitaan, koska on mahdotonta ennustaa, kuinka pelaaja on vuorovaikutuksessa videopelin kanssa. (Slick 2017.)

Esirenderöintiä käytetään yleisesti tilanteissa, joissa nopean renderöinnin sijaan käytetään moniytimisiä suorittimia. Tätä tyyppiä käytetään etenkin animaatioiden ja tehosteiden renderöinnissä, jotka ovat visuaalisesti rikkaita. (Slick 2017.)

Renderöintiin käytetään kolmea eri tekniikkaa, joilla jokaisella on hyvät sekä huonot puolensa: Scanline, Raytracing ja Radiosity. Radiosity- ja raytracing-renderöintejä käytetään yleensä yhdessä, jotta saavutettaisiin mahdollisimman realistista lopputulosta. (Slick 2017.)

Scanline-renderöintiä käytetään esimerkiksi videopeleissä, kun tärkeä tekijä on renderöinnin nopeus. Tällä tekniikalla lasketaan jokaisen objektin pinta kerrallaan. Raytracing-renderöinnissä lasketaan jokainen pikseli kerrallaan käyttämällä mallien pinnoista kimmonneita valonsäteitä, jotka välittyvät 3D-ympäristössä käytettävään kameraan. Verrattuna scanline-renderöintiin tämä tekniikka on hitaampi. Kolmantena menetelmänä radiosity-menetelmä, joka suorittaa laskennat kamerasta riippumatta. Ensisijaisena tarkoituksena on simuloida mallien pintavärejä käyttämällä epäsuoraa valaistusta. (Slick 2017.)

2.4.2 Renderöintimoottorit

Renderöintisovellukset laskevat suuria, pitkälle vietyjä laskelmia. Nykyisillä sovelluksilla käyttäjien ei tarvitse lähestyä taustalla toimiviin laskentoihin, sillä käyttäjillä on käytössä eri parametreja renderöinnin asetuksien muuttamiseen. (Slick 2017.)

Blender-sovelluksessa yhtenä renderöintimoottorina käytetään Cycles Render Engine. Se on raytracing-renderöintiä hyödyntävä renderöintimoottori. Apache 2.0 -lisenssin alla julkaistu renderöintimoottori on avointa lähdekoodia. (Cycles n.d.)

2.5 Tiedoston vienti

2.5.1 Yleistä

Blender-sovelluksesta objekti voidaan viedä tallentamalla se käyttämällä haluttua tiedostonmuotoa. Esimerkiksi malli voidaan viedä toiseen sovellukseen muokattavaksi. Blender tukee laajasti eri tiedostonmuotoja, joita voidaan käyttää sekä viemisessä että tuomisessa. Näitä ovat muun muassa STL, FBX, DAE ja ABC. (Importing and Exporting Files n.d.)

STL-tiedostonmuotoa käytetään silloin, kun objekti viedään CAD-sovellukseen. Tätä tiedostonmuotoa on yleisesti käytössä 3D-tulostussovelluksissa. FBX- ja DAE-tiedostonmuotoja käytetään videopelien kehityksessä, etenkin animoitujen mallien viemiseen. Viimeisenä ABC-tiedostonmuotoa käytetään, jos halutaan viedä tai tuoda suuria määriä tietoa. (Importing and Exporting Files n.d.)

2.5.2 FBX-tiedosto

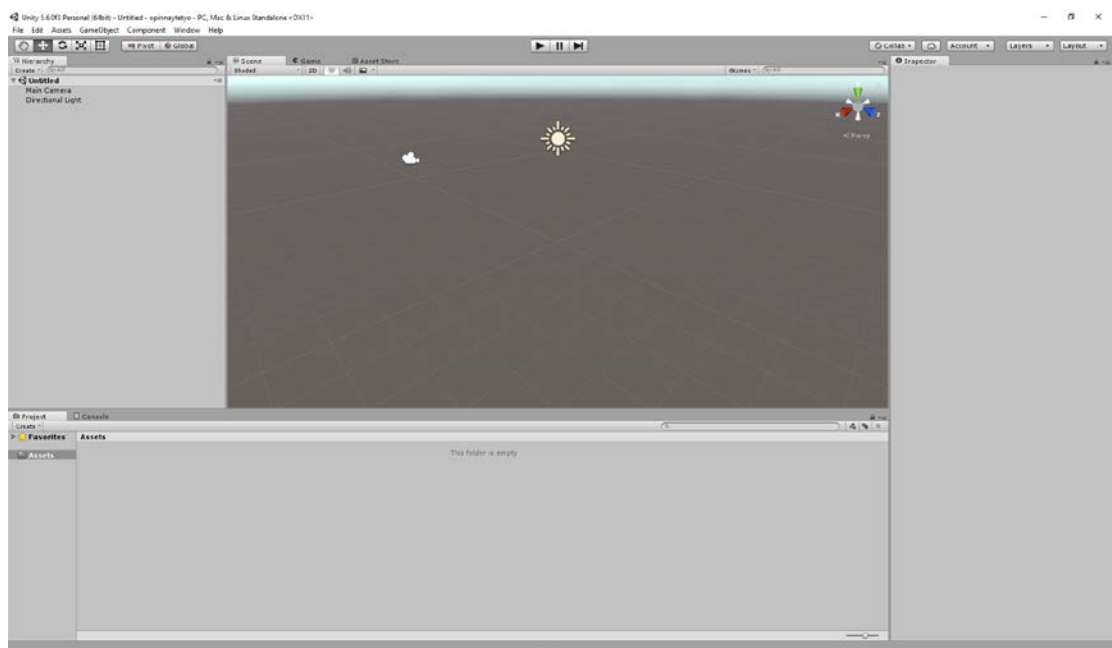
Kun FBX-tiedostoa viedään 3D-sovelluksesta Unity3D-sovellukseen, kannattaa miettiä mitä vietyyn tiedostoon haluaa tallentaa. Blender-sovelluksen tiedoston vientinäköymässä voidaan valita, viedäänkö valittu elementti (esimerkiksi objekti tai animaatio) tai kaikki Blender-tiedostossa olevat elementit. Hyvä työskentelytapa on pitää kaikki elementit Blender-tiedostossa, mutta viedessä mallia Unity3D-sovellukseen vietäisiin ainoastaan haluttu, valittu elementti. (FBX export guide n.d.)

Ennen mallin viemistä FBX-tiedostoksi on hyvä valmistella mallit. Mallien modifierit olisi hyvä hyväksyä ennen vientiä. (FBX export guide n.d.) Lisäksi vietävä mallin kierto

ja skaalaus olisi hyvä hyväksyä käyttämällä Apply Rotation & Scale -valintaa Object Mode -näkyvässä. Hyväksyessä mallin kierron ja skaalauksen Blender-sovellus olettaa kierron olevan nolla astetta jokaisella akselilla (X, Y ja Z), ja skaalauksen asettaa vakioksi jokaisella akselilla. (Reset Object Transformations n.d.)

3 Unity3D

Unity3D (myöhemmin Unity) on Unity Technologies kehittämä kehitystyökalu, jonka avulla kehittäjät voivat toteuttaa kaksi- ja kolmeulotteisia pelejä sekä sovelluksia (ks. kuvio 13). SourceDNA:n tekemän tutkimuksen mukaan 1000 ladatuimmasta mobiili-peleistä 34 % on kehitetty Unity-pelimoottorilla. Unity Technologies asiakaskuntaan kuuluvat muun muassa Coca-Cola, Disney, Electronic Arts ja Square Enix. Unity-kehitystyökalu tukee monipuolisesti eri alustoja kuten Android, Playstation 4 ja Nintendo Switch (The leading global game industry software n.d.)



Kuvio 13. Ruudunkaappaus Unity-kehitystyökalun käyttöliittymästä

Unity-kehitystyökalusta on olemassa neljä erilaista pakettia, joista Personal on ladattavissa ilmaiseksi. Ilmaisuuden ja rojaltivapauden vuoksi aloittelijat sekä harrastajilla on mahdollisuus saada kaikki Unity-sovelluksen ominaisuudet käyttöön. (Unity Personal n.d.)

Personal-versiota voidaan käyttää kaupalliseen kehitykseen, jos vuotuinen tulo on vähemmän kuin 100 000 dollaria tilikautta kohden. Personal-version lisäksi Unity-kehitystyökalusta löytyy Plus-, Pro- ja Enterprise-versiot (Welcome to Unity n.d; Unity Personal n.d.)

3.1 Objektin lisääminen

Unity tukee oletuksena Blender-tiedostoja (tiedostonmuoto .blend). Tallentaessa tiedoston Unity-projektin Assets-kansioon, Unity osaa käyttämällä taustalla toimivaa Blender FBX viejää lukemaan ja lataamaan mallit käytettäväksi. (Importing Objects From Blender n.d.) Lisäksi Unity tukee FBX-tiedostoja, joita voi generoida monella paljon käytetyillä 3D-sovelluksilla (FBX export guide n.d.).

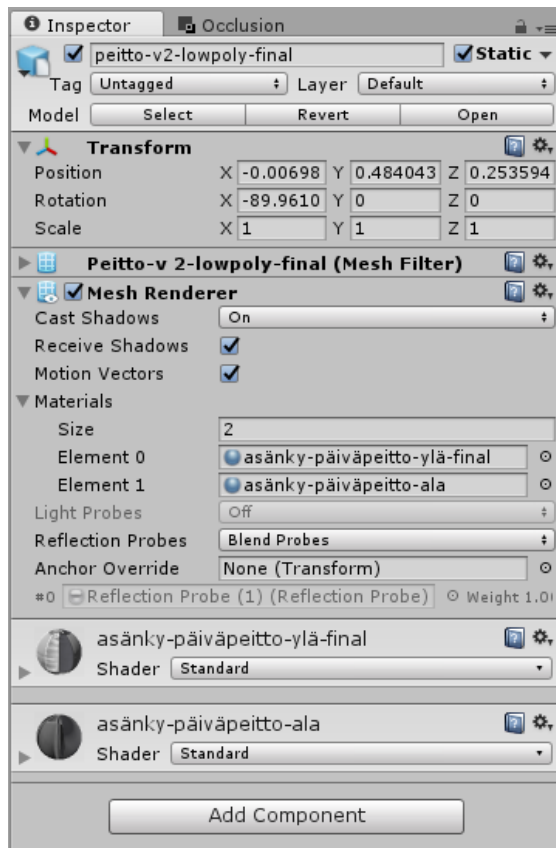
Unity tukee malleja (englanniksi assets) jotka on tuotu 3D-mallintamissovelluksista vietyinä FBX-, DAE-, 2DS-, DXF- tai OBJ-tiedostoon. Malleja voidaan tuoda Unity-sovellukseen käyttämällä Import New Asset -työkalua. (Felicia 2013, 40-41.)

3.2 Objektin asetukset

3.2.1 Yleistä

Inspector-ikkunan avulla voidaan muokata jokaisen kehitettävän pelin objektin asetuksia. Valittaessa minkä elementin tahansa sen asetukset avautuvat kyseiseen ikkunaan. (Goldstone 2009, luku 1.) Inspector-ikkunaan avautuu kaikki komponentin osat, joita voidaan muokata käyttämällä yksinkertaista lomakerakennetta (tekstilaatikot, liikusäätimet, valintalaatikot).

Kuviossa 14 nähdään objektin "peitto-v2-lowpoly-final" tiedot Inspector-ikkunassa. Transform-osiossa voidaan muuttaa objektin sijaintia, kiertoa ja skaalaa.



Kuvio 14. Inspector-ikkuna

3.2.2 Materiaalit

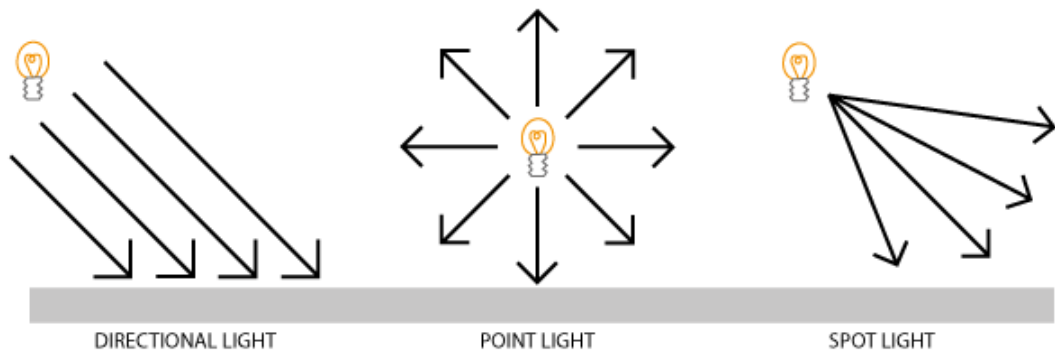
Materiaalit ovat 3D-sovelluksissa yleinen konsepti, joilla tuodaan visuaalinen ulkomuoto 3D-malleille. Materiaalit käsittelevät kaikkea tavallisesta väristä aina kiiltävään pintaiseen materiaaliin. (Goldstone 2009, luku 1.)

Materiaali toimii shaderin kanssa, joka on skripti mikä vastaa renderöinnin tyylistä. Esimerkiksi heijastava shaderissa materiaali renderöi mallin ympärillä olevien objektien heijastukset, kuitenkin säilyttäen objektin värin tai asetetun kuvatekstuurin. (Goldstone 2009, luku 1.)

Unity-sovelluksessa tiedoston tuomisessa (import) siirtyy myös materiaalit, jotka on luotu toisessa 3D-mallinnussovelluksessa. Tästä Unity osaa luoda automaattisesti asettin myöhempää käyttöä varten. Materiaali voidaan myös luoda tyhjästä. Tällöin materiaalille voidaan asettaa tekstuurit tai värin. Unity-sovelluksessa tulee mukana kattava shader-valikoima, mutta käyttäjä voi luoda oman shader-skriptin. (Goldstone 2009, luku 1.)

3.3 Valaistus

Jotta voimme nähdä objektit, Unity-sovelluksesta löytyy kolme valotyyppiä: Directional-valo, Point-valo sekä Spot-valo (ks. kuvio 15). Directional-valoa käytetään päävalolähteenä, kuten auringon valona. Point-valo on lähtöisin yhdestä pisteestä ja sitä käytetään esimerkiksi sisävaloina, tulessa sekä hehkuissa malleissa. Spot-valo lähettää valoa valittuun suuntaan, mutta valolla on taskulampun tavoin alue, jonka valo valaisee. (Goldstone 2009, luku 2.)



Kuvio 15. Unity-sovelluksen kolme valotyyppiä

Valaistus voidaan tallentaa tekstuuriin käyttämällä Create Lightmap -työkalua. Työkalulla voidaan tallentaa monen valon vaikutukset kuten esimerkiksi auringon (Directional-valo) ja ulkovalaisimet (Point-valo). Unity-sovelluksen käyttäessä lightmap-kuvia saadaan säästettyä laskentatehoja, sillä osa valaistuksesta on jo valmiiksi laskettu tekstuureihin. (Goldstone 2009, luku 2.)

4 CASE: Asuntoesittelysovellus

Opinnäytetyössä keskityttiin toteuttamaan sängyn mallintaminen suunnittelusta mallien toteuttamiseen sekä mallinnetun objektin sijoittamisesta sovellukseen.

Sänky-objektin lisäksi opinnäytetyössä keskityttiin huoneiston valaistuksen toteuttamiseen. Valaistus oli toteutettu jo osittain valmiiksi asuntoesittelysovellukseen, mutta Zaibatsu halusi kehittää valaistusta valoisammaksi sekä realistisemmaksi.

4.1 Sängyn toteuttaminen

Zaibatsun asiakas halusi toisen huoneiston suuremmasta huoneesta aikuisten makuuhuoneen. Huoneeseen tarkoituksena oli asettaa kahden hengen sänky, yöpöydät, kaapisto ja koriste-esineinä jalkalamppu ja seinälle tauluja.

4.1.1 Lähtökohdat ja suunnittelu

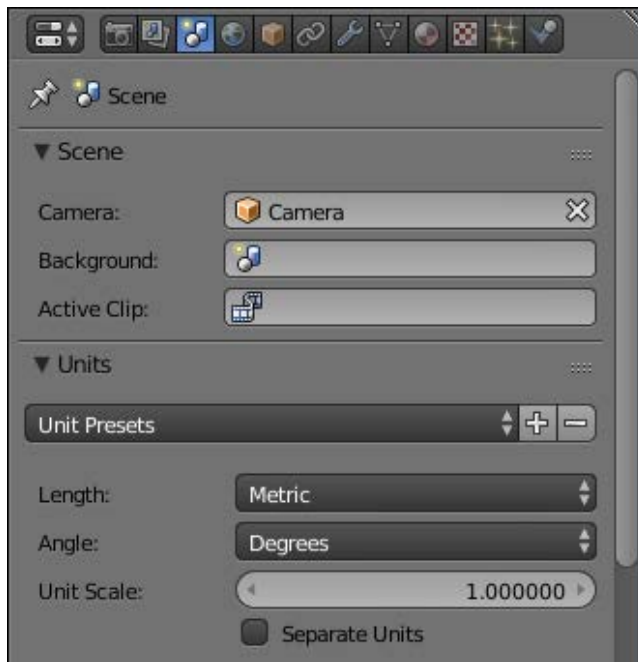
Objektin suunnittelu alkoi kysymyksillä mitä, millainen sekä minne. Sänky tulitisiin sijoittamaan aikuisten makuuhuoneeseen. Zaibatsun asiakas toivoi kahden hengen sänkyä, minkä päiväpeitto olisi taiteltu sängyn jalkapäädyistä.

Suunnittelussa tärkeintä on tutkia sen hetkistä tarjontaa verkossa. Tässä tapauksessa suunnittelun aikana tutkittiin erilaisia sänkymalleja käyttämällä Pinterest ja Google-hakupalvelua. Tuloksiksi saatiin jenkkisänkyjä päädyillä sekä ilman päätyjä, runkopatjoja sekä perinteisempiä sänkymalleja. Näistä malleista valittiin jenkkisänky.

Sängyn leveydeksi valittiin 1.6m ja syvyydeksi 2.0m. Sängyn rungonpatjan paksuudeksi valittiin 0.23m, joustinpatjan paksuudeksi 0.18m ja sijauspatjan paksuudeksi 0.08m.

4.1.2 Mallintaminen

Suunnittelun jälkeen siirryttiin mallintamiseen. Ennen mallintamisen aloittamista on tärkeää tarkastaa Blender-mallinnusohjelman asetukset. Malli tehtiin mittojen mukaan, joten mittayksiköksi valittiin metrit ikkunassa Units (Properties > Scene > Units) (ks. kuvio 16).



Kuvio 16. Näyttökaappaus Blender-sovelluksen Units-näkymästä.

Patjojen ja sängyn jalkojen toteuttaminen

Mallintaminen aloitettiin yksinkertaisilla elementeillä kuten patjoilla ja sängyn jaloilla. Patjat ja sängyn jalat olivat yksinkertaisia kuutioita, jotka pyöristettiin kulmista. Pyöristämällä kulmat käyttämällä yhtä tai kahta ylimääräistä pintaa saatiin realistisemmän näköinen muoto. Käyttämällä Bevel-työkalua saatiin nopeasti luotua halutun laisen pyöristyksen kulmiin.

Sängyn tyynyt ja päiväpeitto vaativat sekä korkea- että matalaresoluutioiset mallit. Korkearesoluutioisilla malleilla saatiin luotua muotoja, jotka voitiin tallentaa normal map -tekstuuriin, jonka avulla tekstuuriin tallennetut yksityiskohdat saatiin näkyväksi matalaresoluutiossa mallissa. Matalaresoluutio mallissa pintojen vähäisy vaikeuttasi luonnollisten muotojen kuten kankaan ryppyjen mallintamisen.

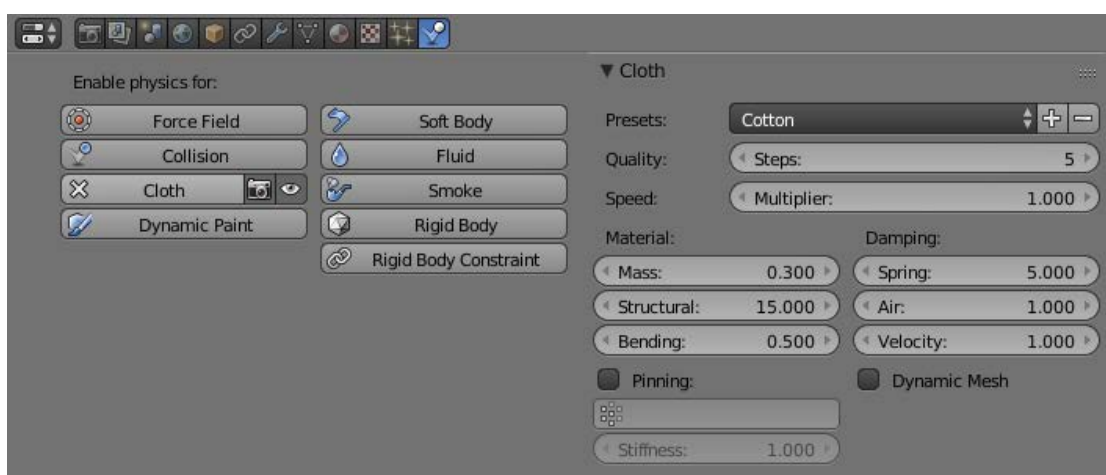
Objektit UV-kartoitettiin, jotta niihin voitiin asettaa myöhemmin tekstuurit. Laatikko-mallisissa patjoissa sivuihin asetettiin Mark Seam -reunat, helpottamaan teksturointia.

Tyynyn toteuttaminen

Tyyny voidaan luoda monella eri tavalla kuten esimerkiksi liikuttamalla pintoja Edit Mode -näkyvässä, jotta objektista saataisiin tyynyn mallinen. Yksi menetelmä on käyttää Blender-sovelluksen fysiikkamoottoria. Ennen fysiikkamoottorin käyttämistä

luotiin ohut suorakulmainen särmiö. Mallinnettavaan eli korkearesoluutioiseen tyy-nyyn asetettiin 8468 pintaa, sillä mitä enemmän mallissa on pintoja, sitä realistisemman näköinen tyynty saadaan toteutettua.

Luodun suorakulmaisen särmiön keskipisteeseen asetettiin Force-objekti, jonka avulla saatiin objekti laajenemaan jokaiseen suuntaan. Tyynty-objektiin asetettiin Cloth-fysiikka, jonka avulla objekti mukautui siihen kohdistuviin voimiin halutun kangasmateriaalin ominaisuuksien mukaan. Kangasmateriaaliksi valittiin Blender-sovel- luksessa valmiiksi löytyvän Cotton-materiaali. Kankaan materiaalin voi myös itse aset- taa haluaman laisekseen muuttamalla arvoja Physics-ikkunassa (ks. kuvio 17).



Kuvio 17. Näyttökaappaus Blender-sovelluksen Physics-ikkunan Cloth-näkymästä.

Ennen fysiikoiden laskemista tyynty-objektista kannatti poistaa massa, jotta tyynty ei putoa alaspäin ajaessa fysiikkamoottoria. Blenderissä gravitaatio on vakiona kytket- tynä päällä, jolloin tyynty tippuisi ja tyyntyn keskipisteeseen keskitetty voima ("Force") joutuisi tyyntyn putoamisen takia tyyntyn ulkopuolelle. Tällöin mallin muotoutuminen tyyntyksi epäonnistuu, sillä voima ei kohdistu enää kappaleen sisältä työntäen pintoja ulospäin.

Käyttämällä toisto-ominaisuutta (englanniksi "Play render animation") saatiin rende- röityä animaatio. Muuttamalla voiman arvoa Strength saatiin muutettua kappaleen muotoutumisen nopeutta. Jos tyynty muotoutui voiman vaikutuksesta liian hitaasti, arvoa kasvattamalla muotoutumista saatiin nopeutettua. Mitä korkeampi voiman arvo on, sitä nopeammin kappaleesta muotoutuu tyynty.

Kun tyyny muotoutui halutun laiseksi, tyynyn muoto hyväksyttiin Modifiers-ikkunassa. Saatua mallia kutsuttiin high poly eli korkearesoluutioiseksi malliksi sen pintojen lukumäärän takia. Tämän jälkeen tyyny oli uudelleen muokattavissa. Skaalaamalla tyynyn sivuja sisälle päin saatiin tyynyn ympärille luotua saumamainen rakenne.

Tyynyyn saattoi jäädä kohtia, joissa pinta muodostui huonosti. Tällöin pinnalla saattoi esiintyä teräviä kulmia tai silmiin erottuvia, neliön muotoisia pintoja. Nämä saatiin pois käyttämällä Sculpt Mode-näkymän Smooth-työkalua.

Pitääkseen pintojen lukumäärää pienenä tyynyn pohjasta poistettiin pintoja. Tyynyjä ei käytetty muiden huonekalujen yhteydessä kuin sänky-objektissa, joten käyttäjältä piilossa olevat pinnat tyynyn pohjasta poistettiin.

Tyynyn valmiiksi saamisen jälkeen tyynylle luotiin tekstuuri UV-kartoituksella. Aiemmin luodun saumamaisen rakenteen kohdalle asetettiin sauma Mark Seam-työkalulla. Käyttämällä Unwrap-työkalua tyynylle saatiin asetettua tekstuuri.

Low poly -mallin toteuttaminen voidaan tehdä monella eri tavalla. Tyyny-objektin yksinkertaisen mallin ja vähäisen pintamäärän takia low poly -malli tehtiin häivyttämällä ylimääräisiä reunoja manuaalisesti. Tämä tapahtui valitsemalla halutut poistettavat reunat ja käyttämällä Edge Loop -toimintoa, jolloin Blender valitsee aina seuraavan reunan muodostaen silmukkamaisen valinnan. Joskus Edge Loop -toiminto ei toiminut halutulla tavalla, jolloin reunat valittiin sekä häivytettiin yksitellen. Häivytyksessä (englanniksi dissolve) Blender poistaa valitun reunan yhdistäen kaksi tasoa yhdeksi tasoksi. Tällöin on hyvä muistaa pitää topologia kunnossa, ettei pinnan muoto muutu radikaalisesti high poly -mallista.

Käyttämällä high poly ja low poly versiota saatiin high poly -mallin pinnanmuodot tallennettua tekstuuriin käyttämällä Case-menetelmää. Tallennettua kuvatiedostoa käytettiin myöhemmin Unity-sovelluksessa normal-kuvana.



Kuvio 18. Tyyny-mallit sekä pintojen lukumäärät.

Kuviossa 18 nähdään tyyny-mallin vaiheet Blender-sovelluksessa vasemmalta oikealle editointi-näkymässä sekä alla renderöitynä: ennen tyynyn muotoilua käyttämällä voimaa, high poly -tyyny ja low poly -tyyny. Tyynyjen yläpuolella ilmoitettu pintojen lukumäärä.

Sängyn päiväpeiton toteuttaminen

Sängyn päiväpeitto toteutettiin käyttämällä samaa Cloth-fysiikkaa, minkä avulla myös tyyny toteutettiin. Ennen Cloth-fysiikkaa peiton leveydeksi asetettiin 2.7m ja pituudeksi 2.5m.

Sängyn päädyn puolella peitto taiteltiin kahteen osaan asiakkaan toiveiden mukaisesti. Tämä tapahtui valitsemalla Edit Mode -näkymässä haluttu alue, joka käännettiin Rotate-työkalua käyttämällä. Peitto taitettiin muuttamatta pintojen muotoja valmiiksi ennen fysiikoiden asettamista.

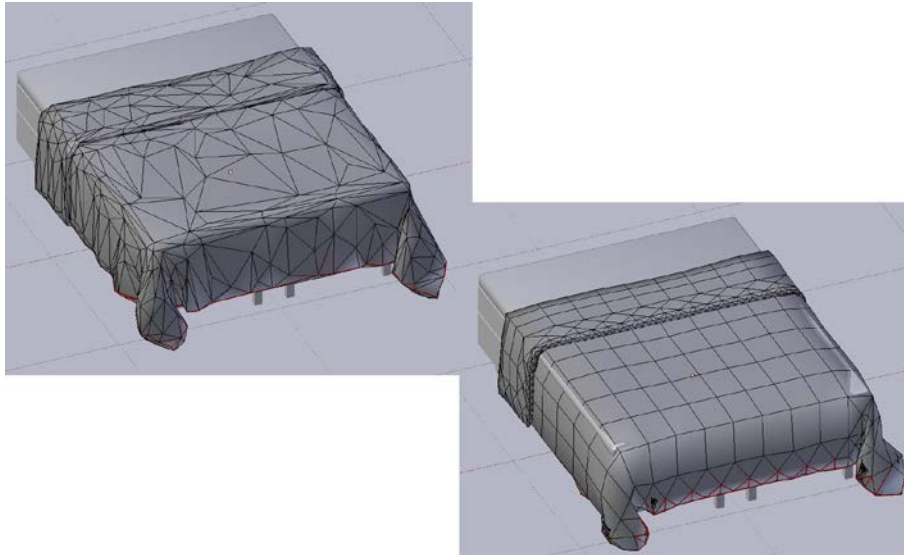
Aiemmin mallinnettuihin patjoihin asetettiin Collision-fysiikat, jotta peittoon asetettu Cloth-fysiikka ottaisi kontaktia patjoihin ja muotoutuisi niiden mukaisesti. Lisäksi sängyn alle lisätylle levyille asetettiin Collision-fysiikka, minkä avulla jäljiteltiin lattiaa. Levyn avulla peiton asettuessa sängyn päälle sen laidat mukautuisi lattian pinnan mukaisesti.

Peitolle asetettiin Cotton-valmismateriaalin fysiikka-asetukset, jota myös käytettiin tyynyjen mallintamisessa. Cloth-fysiikan asetuksissa jouduttiin muuttamaan arvoja, jotta kangas käyttäytyisi oikein mukautuessaan patjan pintojen mukaan. Kankaan venymistä sekä taittumista jouduttiin fysiikkamoottorilla ajamaan useaan otteeseen, jotta kangas ei venyisi liikaa ja taittuisi sängyn päädyn puolelta halutun laisesti.

Animaation renderöinnin sekä mallin hyväksymisen jälkeen päiväpeitolle määritettiin paksuus Solidify-työkalulla. Päiväpeiton paksuudeksi asetettiin lopulta eri vaihtoehtojen tarkastelun jälkeen 0.01m.

Ennen low poly -mallin luomista peitosta poistettiin ylimääräiset pinnat peiton pohjasta, jotka olivat tyynyjen tapaan käyttäjältä piilossa. Tämän jälkeen mallille tehtiin UV-kartoitus, jotta peitolle saatiin toteutettua normal map-kuva. Käyttämällä Mark Seam -työkalua päiväpeitto paloitettiin kahteen osaan, koska pinnan monimutkikkaan muodon tallentaminen kuvatiedostoon onnistuisi helpommin. Lisäksi kahden tekstuurin hyödyntäminen päiväpeitossa mahdollisti erilaisten tekstuurien asettamisen päiväpeiton päällimmäiselle sekä pohjimmaiselle pinnoille. Tässä kohtaan pintojen lukumääräksi Blender-sovellus ilmoitti 296,416 pintaa.

Seuraavana vuorossa oli low poly -mallin luominen, joka muodostui haastavaksi. Ensimmäisenä menetelmänä mallilta yritettiin poistaa ylimääräisiä pintoja käyttämällä Decimate-työkalua, minkä avulla low poly -version luominen olisi ollut nopeampaa. Pintojen lukumäärä saatiin tiputettua alkuperäisestä, noin 296 000 pinnasta alle 4000 pintaan, mutta mallien topologia kärsi. Työkalun Collapse-ominaisuudella mallista saatiin saman muotoinen kuin alkuperäinen high poly -malli, mutta säännöllinen verkkomainen rakenne muuttui reunoista kolmion muotoisten pintojen sekamelskaksi (ks. kuvio 19.). Työkalun Un-Subdivide-ominaisuudella saatiin pidettyä verkkomainen rakenne, mutta tiputtaessa työkalulla pintojen lukumäärää mallin muoto muuttui kulmikkaaksi. Helmojen taitokset menettivät pehmeät, kankaanmaiset linjat ja reunoihin ilmestyi kolmion mallisia pintoja (ks. kuvio 19). Kummassakin tapauksessa aiemmin UV-kartoituksen aikana asetetut Mark Seam -reunat muuttuivat jolloin mallin UV-kartoitus ei ollut enää yhtäläinen high poly -mallin kanssa.



Kuvio 19. Decimate-työkalun Collapse (vasemmalla) ja Un-Subdivide (oikealla).

Low poly -mallin toteutettiin lopulta häivyttämällä halutut reunat Edit Mode - näkymässä Dissolve Edges -valinnalla. Reunoja poistettiin siten, ettei mallin muodot kärsineet. Lopulta pintojen lukumääräksi saatiin 2 096.

Normal-kuvan luominen Bake-työkalulla oli haastavaa. Verrattuna tyynyn normal map -kuvan tallentamiseen peitossa oli paljon muotoja etenkin helmoissa, jotka vaikeuttivat tekstuurin tekoa. Monen yrityksen jälkeen normal map onnistuttiin tekemään sekä päällimmäiselle- että pohjapinnalle.

Lopulta sängyn mallintaminen saatiin valmiiksi. High-poly -malleista luodut low-poly -mallien yhteispintamääräksi saatiin 2 982 pintaa (ks. kuvio 20).



Kuvio 20. Valmiiksi saatu sänky-malli Blender-sovelluksessa renderöitynä.

4.1.3 Mallin vieminen tiedostoon

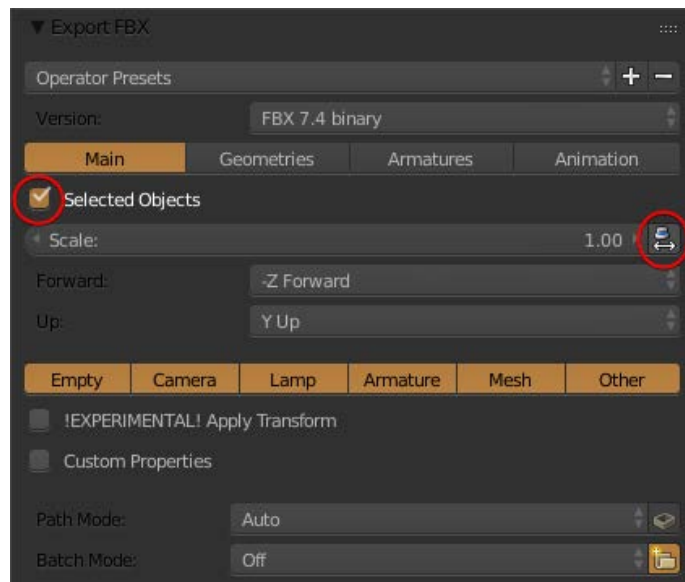
Mallien valmiiksi saamisen jälkeen mallit vietiin tiedostoon. Tiedostomuodoksi valittiin FBX, jota Unity-kehitystyökalu tukee.

Ennen mallien viemistä tiedostoihin tarkastettiin vietävien mallien kierto (Rotation) ja skaala (Scale). Hyväksymällä Apply-ponnahdusikkunalla objektien kierrot ja skaalaukset saatiin ennaltaehkäistyä ongelmia Unity-kehitystyökalun puolella.

Vienti-näkymässä vientiasetuksissa valittiin Selected Objects sekä poistettiin valinta kohdasta Scale. Selected Objects -valinnan avulla saatiin vietyä juuri ne mallit, mitkä haluttiin viedä.

Unityssa monen asuntoesittelysovellukseen mallinnettujen objektien viemisestä tuli ongelmia, kun Scale-ominaisuus oli kytkettynä vientiasetuksissa. Poistamalla valinta kohdasta Scale skaalaus Unity-kehitystyökalun puolella saadaan pidettynä vakiona. Jos valintaa ei poista Scale-kohdasta, skaalauksen luku Unityssa kertaantuu 100. Kuitenkin kappaleen koko pysyy samana Unityssa, oli Scale-ominaisuus valittuna tai ei.

Kuviossa 21 vientiasetusten ikkuna. Punaisilla ympyröillä merkittynä Selected Objects -valinta sekä poistettu valinta Scale.



Kuvio 21. Mallien vientiasetukset.

4.1.4 Objektin lisääminen asuntoesittelysovellukseen

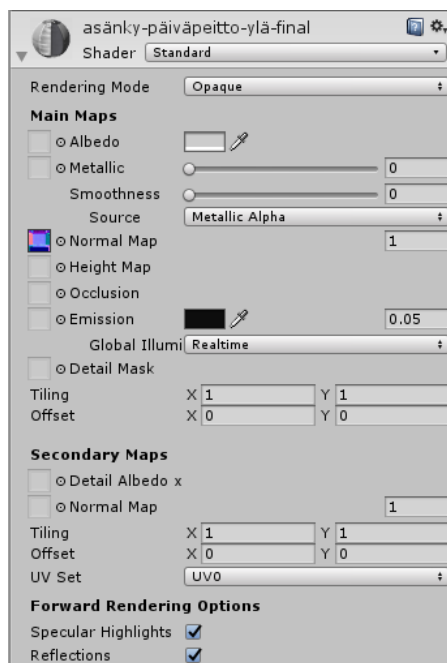
Sänky lisättiin sovellukseen import-työkalua käyttämällä. Malli sijoitettiin paikalle, mihin asiakas oli suunnitellut sängyn laitettavan. Jos Blender-sovelluksessa ei hyväksynyt objektin - tässä tapauksessa sängyn - kierrettä ja skaalausta, nämä arvot olisivat siirtyneet mukana Unity-sovellukseen objektin kiertoon ja skaalaukseen.

Sängyn mukana saattoi tulla myös muita elementtejä, joiden ei ollut tarkoitus viedä Blender-sovelluksesta Unity-sovellukseen. Tällöin valinta Selected Objects Export FBX -ikkunassa oli unohtunut.

4.1.5 Materiaalin asettaminen

Sängyn tekstuurit ladattiin Unity-sovellukseen ja asetettiin objekteissa oleviin materiaaleihin. Myös normal map -tekstuurit asetettiin sille tarkoitettuun kohtaan Normal Map. Muuttamalla normal map -arvoa (vakiona 1) pystyttiin vaikuttamaan siihen, kuinka voimakkaasti tekstuuri vaikuttaa.

Kuviossa 22 on Unity-sovellukseen tuodun mallin ”asänky-päiväpeitto-ylä-final” materiaali. Albedo-väriksi on valittu harmaa, joka on mallin väri. Normal map -kohtaan on sijoitettu Blender-sovelluksella toteutettu normal map -kuva, jotta saadaan low-poly



Kuvio 22. Materiaalin ominaisuuksia Unityssa

Joissakin objekteissa käytettiin netistä ladattuja tekstuureja kuten tyynyissä, patjoissa ja sängynjaloissa. Nämä tekstuurit ladattiin palvelusta Textures.com, joka tarjoaa digitaalisia kuvia kaikenlaisista materiaaleista kuten kankaista, puista, metalleista ja muoveista (About n.d.). Peitolle ei asetettu tekstuuria, kuitenkin se on mahdollista asettamalla tekstuuri Albedo-kenttään.

4.2 Valaistus

Valaistuksen toteutuksessa otettiin huomioon monia eri tekijöitä kuten sen hetkiset valoa loistavat objektit (emission-materiaali). Ennen valaistuksen muuttamista huoneistojen ainoat valonlähteet olivat peitetyt ikkunat. Lisäksi peittämällä ikkunat esitettiin pelaajien näkeminen ulos ikkunoista, sillä pelaajan haluttiin ainoastaan keskittyvän huoneistoihin.

Aluksi Unity-skeneen toteutettiin aurinko käyttämällä *directional*-valoa. Valo asetettiin siten, että sen tuottama valo tulisi mahdollisimman monen huoneen sisälle ikkunoiden kautta. Voimakkuutta säätämällä saatiin valolla valaistua huoneita, mutta nostamalla valon voimakkuutta liian korkealle osa huoneista ylivalottuivat. Kuviossa 23 nähdään *directional*-valon vaikutus asuntoesittelysovelluksen huoneiston keittiössä.

Käyttämällä *Final Gather* -ominaisuutta saatiin keittiökalusteiden pinnoille luotua realistinen valonkaje. *Final Gather* -ominaisuudella saadaan parannettua *lightmapin* visuaalista laatua samalla lisäen *lightmap*-tekstuurien bakettamisen kestoa (Lighting Window 2017).

Vaikka *Final Gather* -ominaisuuden avulla saatiin parannettua sovelluksen visuaalista ilmettä, sen käyttämisestä syntyi ongelmia. Valon heijastuessa kiiltäväpintaisista materiaaleista seiniin muodostui valoalueita. Myös tiloissa, joissa valaistusten puutteen vuoksi oli pimeämpää, seinille muodostui valoalueita. Ongelma saatiin ratkaistua poistamalla valoa heijastavien objekteista kuten pesuhuoneen lasiseinästä *Reflections*-ominaisuus, jolloin kappale ei enää heijastanut valoa. Ominaisuuden poiskytkeminen muutti kappaleen väritystä, sillä kappale ei enää heijastanut sen ympärillä olevien mallien materiaaleja.

Kuviossa nähdään myös muita kalusteita asuntoesittelysovellukseen opinnäytetyön ohessa mallinnettuja kalusteita kuten keittiön kaapisto, sähköhella, astianpesukone ja liesituuletin.



Kuvio 23. Asuntoesittelysovelluksen keittiö

Peitetyt ikkunat, jotka valaisivat huoneen aiemmin, jätettiin paikoilleen (ks. kuvio 23). Ikkunan ominaisuuksia muutettiin, jotta directional-valo saataisiin kulkemaan ikkunaobjektien läpi huoneistoihin. Tämä saatiin toteutettua ottamalla ominaisuus Cast Shadows pois käytöstä valituilta objekteilta.

Realistisen auringonvalon (directional-valo) ja ikkunoiden (emission materiaali) materiaalin arvoja muutettiin useaan otteeseen, jotta huoneistojen valoisuus ei ollut liian kirkasta eikä liian pimeää. Kun suuremman huoneen valaistus saatiin halutun laiseksi, vaikutti se myös muihin huoneisiin. Tällöin toisista pienemmistä huoneista saattoi tulla ylivalottuneita ja toisista huoneista alivalottuneita. Tämä saatiin ratkaistua osittain asettamalla ikkuna-objekteihin eri materiaaleja, jotka erosivat hieman toisistaan. Huoneissa, joissa ylivalottuvuus oli ongelma, muutettiin emission-arvoa pienemmäksi, kun taas alivalottuneiden huoneiden ikkuna-materiaalien emission-arvoa nostettiin.

Vaikka huoneistoja valaistiin käyttämällä directional-valoa sekä emission-valoja, osa huoneista jäi pimeäksi. Tällöin asuntoesittelysovelluksen käyttäjälle saattoi välittyä olettaus, ettei huoneistot ole valoisia. Tämä lopulta korjattiin lisäämällä seinä- ja

kattoelementteihin emission-ominaisuus. Jo pienillä arvoilla huoneistoista saatiin valoiskempia (ks. kuvio 24). Myös joidenkin sovelluksen mallien materiaaleihin asetettiin emission-valo. Jo pienillä arvoilla materiaaleista saatiin kirkkaampia ja värikkäitä.

Kuviossa 24 nähdään valaistuksen lisäksi muita opinnäytetyön ohessa mallinnettuja objekteja kuten suihkuseinä, kattovalaisin ja WC-istuin.



Kuvio 24. Toinen huoneisto valaistuksen jälkeen

5 Tulokset ja pohdinta

Opinnäytetyön tavoitteena oli tutkia 3D-mallintamista sekä valaistuksen asettamista Unity-pelimoottorissa. Lopputuloksena asuntoesittelysovellus sai valoisan ja realistisen valaistuksen sekä sängyn aikuisten makuuhuoneeseen. (ks. kuvio 25.)

Olin kiinnostunut mallintamisesta jo ennen ammattikorkeakouluun hakemista. Opinnäytetyön aikana sain paljon lisätietoa ja kokemusta mallintamisesta, objektin viemisestä sekä Unityn ja Blenderin käyttämisestä. Mallintaessa korostui etenkin mallin topologia, jota kannattaa pitää silmällä. Huonosti toteutettu topologia voi Unity-sovelluksessa aiheuttaa kappaleen pinnalle varjoalueita sekä muita valovirheitä. Lisäksi opinnäytetyö osoitti, kuinka kauan jonkun asian toteuttaminen kestää.



Kuvio 25. Mallinnettu sänky sekä säädetty valaistus asuntoesittelysovelluksessa

Materiaaleja tietoperustaan löytyi kattavasti kirjoista sekä internetistä. Suomenkielisten materiaalien vähäisyyden takia suurin osa tietoperustassa käytetyistä materiaaleista olivat vieraskielisiä.

Tekniikan kehittyessä myös sovelluksia kehitetään jatkuvasti eteenpäin. Tällöin pitää miettiä, onko järkevää käyttää vanhoja materiaaleja tietoperustan luomisessa. Tässä tapauksessa vanhojen materiaalien käyttö on perusteltua, koska 3D-mallintamisessa sekä käytetyissä työkaluissa ei ole tapahtunut oleellisia muutoksia viimeisien vuosien aikana.

Sängyn mallintamista sekä valaistuksen päivittämistä tehtiin samanaikaisesti. Sängyn peiton mallintamisen kanssa tuli ongelmia useamman kerran sen monimutkikkaan muotoilun takia. Peitto jouduttiin tekemään alusta alkaen useaan otteeseen, jonka takia sängyn valmiiksi saaminen venyi aina opinnäytetyön viimeisille tunneille saakka. Valaistuksen toteutuksessa aikaa meni eri menetelmien testaamiseen sekä oikeiden arvojen löytämiseen.

Sängyn pinta vaatii mielestäni vielä työstöä, sillä Unityssa sen kulmikkaat muodot tulevat esiin etenkin varjojen bakettamisen jälkeen. Myös valaistus vaatii pientä hienosäätöä, sillä joidenkin ikkunoiden karmit tai muut huoneessa olevat mallit heijastavat valoa, jolloin ikkunoiden pinnoille tai ikkunakarmeihin muodostuu varjo-alueita.

Opinnäytetyön rinnalla työstin asuntoesittelysovellukseen muitakin malleja kuin sänkyä, mikä kartutti osaamista. Näitä malleja olivat muun muassa keittiön kalusteet, tv-taso, pesuhuoneen kalusteet ja hattuhylly.

Työn aikana pidimme kokouksia sekä yhteyttä sähköpostitse Zaibatsun asiakkaiden kanssa. Heiltä saatujen kommenttien perusteella opinnäytetyötä ja asuntoesittelysovellusta vietiin eteenpäin.

Kirjallista osaa käyttämällä lukija pystyy toteuttamaan mallin Blender-sovelluksella aina objektin suunnittelemisesta objektin toteutukseen. Mallin viemisessä täytyy olla tarkkana vientiasetuksien kanssa, jotka voivat vaikuttaa Unityssa objektien asetuksiin ja turhien objektien viemiseen.

Lähteet

About. 2017. Tietoa sivustosta Textures.com. Viitattu 14.5.2017.

<https://www.textures.com/about>.

About. N.d. Tietoa Blender-sovelluksesta. Viitattu 12.3.2017.

<https://www.blender.org/about/>.

Baking Physics Simulations. N.d. Fysiikkasimulaatioiden tallentamisesta. Viitattu 13.5.2017. <https://docs.blender.org/manual/de/dev/physics/baking.html>.

Birn, J. 2014. Digital Lighting & Rendering. 3. painos. Yhdysvallat: New Riders Press.

Blender. License. N.d. Blender-sovelluksen lisensistä. Viitattu 12.3.2017.

<https://www.blender.org/about/license/>.

Clear Object transformations. N.d. Ohjesivu objektin muutoksien nollaamiseen sekä hyväksymiseen. Viitattu 14.5.2017.

https://wiki.blender.org/index.php/User:Fade/Doc:2.6/Manual/3D_interaction/Transform_Control/Reset_Object_Transformations.

Connell, E. 2011. 3D for Graphic Designers. Yhdysvallat: John Wiley & Sons Inc.

Cycles. N.d. Blender-sovelluksen Cycles-renderöintimoottorin johdanto. Viitattu 14.5.2017.

<https://www.blender.org/features/cycles/>.

Derakhshani, D. & Munn, R. 2008. Introducing 3ds Max 2008. Yhdysvallat: Wiley Publishing.

FBX export guide. N.d. Ohjeistus mallin viemisestä 3D-mallinnusohjelmasta Unity-sovellusta silmällä pitäen. Viitattu 14.5.2017.

<https://docs.unity3d.com/Manual/HOWTO-exportFBX.html>.

Felicia, P. 2013. Getting Started with Unity. Iso-Britannia: Packt Publishing Ltd.

Flavell, L. 2010. Beginning Blender. Open Source 3D Modeling, Animation, and Game Design. Yhdysvallat: Apress.

Goldstone, W. 2009. Unity Game Development Essentials. Iso-Britannia: Packt Publishing Ltd.

Gravity. N.d. Blender-sovelluksen fysiikka-moottorin painovoimasta. Viitattu 13.5.2017.

<https://docs.blender.org/manual/de/dev/physics/gravity.html>.

Importing and Exporting Files. N.d. Blender-sovelluksen tiedostojen viennistä sekä tuonnista. Viitattu 14.5.2017.

https://docs.blender.org/manual/en/dev/data_system/files/import_export.html.

Importing Objects From Blender. N.d. Ohjeistus objektien viemisestä Blenderistä

Unityyn. Viitattu 14.5.2017. <https://docs.unity3d.com/Manual/HOWTO-ImportObjectBlender.html>.

Introduction. N.d. Blender-sovelluksen fysiikka-moottorin johdanto. Viitattu 11.5.2017.

<https://docs.blender.org/manual/de/dev/physics/introduction.html>.

- Introduction. N.d. Sculpt Mode -tilan johdanto. Viitattu 13.5.2017.
https://docs.blender.org/manual/en/dev/sculpt_paint/sculpting/introduction.html.
- Introduction. N.d. Cycles Render Enginen johdanto. Viitattu 14.5.2017.
<https://docs.blender.org/manual/en/dev/render/cycles/introduction.html>.
- Kattavat luovat työkalut viihdealan suunnittelua varten. N.d. Autodesk Media- ja Viihdealan Kokoelman kauppa- sekä esittelysivu. Viitattu 4.4.2017. <http://www.autodesk.fi/collections/media-entertainment/overview>.
- License and Services Agreement. N.d. Autodeskin sovelluksien lisensseistä sekä sopimuksista. Viitattu 12.3.2017. <http://download.autodesk.com/us/FY18/Suites/LSA/en-US/lisa.html>.
- Lighting Window. 2017. Ohjeistus Unity-sovelluksen valaistusikkunan käytöstä. Viitattu 23.5.2017. <https://docs.unity3d.com/Manual/GlobalIllumination.html>.
- Murdock, K. 2011. 3ds Max 2012 Bible. Yhdysvallat: John Wiley & Sons Inc.
- Parisi, T. 2014. Programming 3D Applications with HTML5 and WebGL. 3D Animation and Visualization for Web Pages. korj. p. Yhdysvallat: O'Reilly Media.
- Puhakka, A. 2008. 3D-Grafiikka. Helsinki: Talentum Media.
- Slick, J. N.d. Topology in 3D Animation. Topologian määritelmä. Päiv. 20.8.2016. Viitattu 14.5.2017. <https://www.lifewire.com/topology-in-3d-animation-2181>.
- Slick, J. N.d. What is Rendering?. Renderöinnin määrittäminen. Päiv. 4.2.2017. Viitattu 14.5.2017. <https://www.lifewire.com/what-is-rendering-1954>.
- Tools. N.d. Tietoa Sculpt Mode -tilan pensselivalikoimasta. Viitattu 13.5.2017.
https://docs.blender.org/manual/en/dev/sculpt_paint/sculpting/tools.html.
- The leading global game industry software. N.d. Unity Technologies yrityksen kehittämästä Unity-kehitystyökalusta. Viitattu 3.5.2017. <https://unity3d.com/public-relations>.
- Unity Personal. N.d. Unity Personal paketin lataussivu. Viitattu 3.5.2017.
<https://store.unity.com/products/unity-personal>.
- Welcome to Unity. N.d. Unity-kehitystyökalun eri pakettien kauppa- sekä esittelysivu. Viitattu 14.5.2017. <https://store.unity.com/>.
- Zaibatsu Interactive Inc. N.d. Lehdistötiedote Zaibatsun sivustolla. Viitattu 15.2.2017.
<http://zaibatsu.fi/press/>.