

2D-animaation tuottaminen Blenderissä

2D-animointi 3D-ohjelmistolla



Ammattikorkeakoulututkinnon opinnäytetyö

Riihimäki, Mediatekniikan koulutusohjelma

syksy, 2017

Janne Majasalmi

Mediatekniikan koulutusohjelma
Riihimäki

Tekijä	Janne Majasalmi	Vuosi 2017
Työn nimi	2D-animaation tuottaminen Blenderissä	
Työn ohjaaja/t	Antti Laakso	

TIIVISTELMÄ

Tein päättötyöni Riihimäkeläisen drag-artisti Jukka K:n esityksen tausta-animaatioksi. Esitys oli parodia vuoden 2015 euroviisuvoittajan esityksestä, minkä takia animaation ulkoasu pyrittiin kopioimaan mahdollisimman tarkasti euroviisuesityksestä. Animaatio toistettiin isolla led-näytöllä Jukan takana, mikä mahdollisti sen, että animaation hahmosta tuli osa esitystä ja Jukka ja hahmo olivat interaktiivisia. Työstä teki hankalan sen nopea aikataulu sekä minun olematon tietämys animoinnista ja erikoistekniikoilla renderöinnistä. Animaation piti olla valmis kolme viikkoa sen jälkeen, kun keskustelimme aiheesta Jukan kanssa ensimmäistä kertaa. Suurin osa työhön käyttämästäni ajasta kului uusien asioiden opettelemiseen ja tiedon etsimiseen internetistä. Onneksi Blenderille löytyy internetistä paljon laadukkaita opetusvideoita sekä kattava dokumentaatio. Tämän työn tekemisessä aika oli pahin vihollinen ja kolme viikkoa tuntui loppuman auttamatta kesken, eikä animaatiosta tullut paras mahdollinen. Vaikka en lopulta ollut itse täysin tyytyväinen animaation laatuun, ja olisin sitä voinut vielä yrittää parannella, niin asiakas oli tyytyväinen lopputulokseen. Onneksi taustana käytetty led-näyttö oli huonolla resoluutiolla, minkä ansiosta myös hieman isommat virheet eivät olleet selkeästi näkyvissä.

Avainsanat mallintaminen, 3D, 2D, animointi

Sivut 28 sivua

Bachelor of media technology
Riihimäki

Author	Janne Majasalmi	Year 2017
Subject	Producing 2D-animation in Blender	
Supervisors	Antti Laakso	

ABSTRACT

I made my final project for drag artist Jukka K from Riihimäki. It was presented as an background animation in one of his shows which was a parody of eurovision contest winner from 2015. I tried to copy the appearance of animation from eurovision contest as well as I could. The animation was presented on a led-screen behind Jukka which made it possible for Jukka to interact with the character of the animation. Hardest part of the work was really tight schedule and also my lack of experience in animating and rendering with special techniques. I had only three weeks to finish the animation after I talked about it with Jukka for the first time. Most of the time I spent with the work was purely for learning new things and searching for information from the internet. Luckily there is a lot of tutorials and good documentation for Blender. It felt like three weeks wasn't nearly enough for this kind of project but still I managed to finish it. Not in the quality I wished but client was happy with the outcome in the end. Luckily the led-screen they used for presenting the animation was a bit low quality so it didn't show all the flaws so clearly.

Keywords modeling, 2D, 3D, animation

Pages 28 pages

SISÄLLYS

1	JOHDANTO.....	1
1.1	Asiakas.....	1
1.2	Blender	1
2	2D-ANIMAATION TUOTTAMINEN BLENDERISSÄ.....	2
2.1	Freestyle render-moottori	2
2.1.1	Freestylen renderöintiprosessi.....	4
2.2	Freestylen asetukset	7
2.2.1	Viivojen tunnistaminen objektista	7
2.2.2	LineSet	9
2.2.3	LineStyle ja modifierit.....	13
2.3	Mallintaminen	14
2.3.1	Box modeling.....	14
2.3.2	Topologia	15
2.4	Rig ja animointi.....	16
2.4.1	Inverse kinematics.....	17
2.4.2	Keyframe-animointi.....	19
3	2D-ANIMAATION TYÖNKULKU	21
3.1	Mallinnus.....	21
3.1.1	Box modeling.....	21
3.1.2	Face Markien määrittely.....	23
3.2	Riggaaminen.....	23
3.2.1	Rig	23
3.2.2	Inverse Kinematics.....	24
3.3	Animointi.....	26
3.3.1	Ongelmat käsien kanssa	26
3.4	Renderöinti.....	27
	Renderöin lopullisen animaation freestyle-moottoria käyttäen.	Error!
	Bookmark not defined.	

1 JOHDANTO

1.1 Asiakas

Opinnäytetyön soveltava osuus on tehty riihimäkeläiselle viihdetaitelijalle laivalla esitetyn drag show -esityksen taustaksi. Aiheena on Ruotsin vuoden 2015 euroviisuesityksestä tehty parodia. Työn aiheena on tausta-animaatio, joka näytetään esiintyjän taustalla jättiscreenillä ja on vuorovaikutuksessa esiintyjän kanssa. Animaation käsikirjoitus on kokonaan asiakkaan laatima ja graafinen ilme kopioitu Ruotsin euroviisuesityksestä.

Kaksiulotteisesta ulkoasustaan huolimatta animaatio on tehty 3D-työskentelyyn tarkoitetulla Blender-ohjelmistolla. Opinnäytetyön teoriaosuudessa tarkastellaan sitä, miksi olen päätenyt käyttämään 3D-ohjelmistoa 2D-animaation tekemiseen ja mitä se käytännössä tarkoittaa. Lisäksi tarkastelen sitä, mitä asioita on pitänyt ottaa huomioon ja mitä työtekniikoita olen joutunut käyttämään.

1.2 Blender

Blender on ilmainen ja avoimen lähdekoodin 3D-ohjelmisto. Se tukee kaikkia 3D-työskentelyn osa-alueita – mallinnusta, riggausta, animointia, simulointia, renderöintiä, composiointia, motion trackingiä ja jopa videoiden editointia ja pelien tekemistä. Blenderissä on rajapinta Python-scripteille, joiden avulla kokeneet käyttäjät voivat muokata sovellusta ja kehittää omia työkaluja. Usein tällaiset työkalut julkaistaan Blenderin tulevissa versioissa kaikkien käytettäviksi. Blender sopii hyvin yksittäisille käyttäjille ja pienille studioille, jotka hyötyvät sen yhtenäisestä ja tarpeisiin reagoivasta kehitysprosessista.

Blender on cross-platform-sovellus ja se pyörii yhtä hyvin Linuxin, Windowsin ja Macin käyttöjärjestelmillä. Sen käyttöliittymä käyttää OpenGL:ää, mikä takaa yhtenäisen käyttökokemuksen. (Blender Homepage n.d.a)

Blender on julkaistu GNU General Public License (GPL, tai ilmainen ohjelmisto)-lisenssin alla. Tämä lisenssi takaa ihmisille lukuisia vapauksia:

- Blenderiä voi käyttää mihin tarkoitukseen tahansa
- Blenderiä voi jakaa vapaasti
- Blenderin toimintaa voi opiskella ja muuttaa
- Blenderin muutettuja versioita voi jakaa

GPL-lisenssin tarkoitus on suojella tarkoin näitä vapauksia, vaatien kaikkien jakamaan tekemiänsä muutoksia, kun he jakavat ohjelmistoa julkisesti. Tällaiseen näkökulmaan viitataan termillä Copyleft. (Blender Homepage n.d.b)

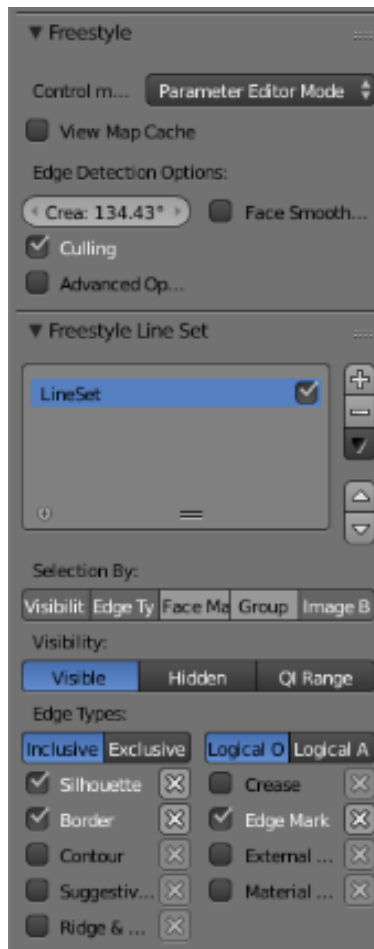
2 2D-ANIMAATION TUOTTAMINEN BLENDERISSÄ

2.1 Freestyle render-moottori

Freestyle render-moottori on ensimmäistä kertaa Blenderin 2.67 versioon integroitu kolmannen osapuolen tuottama lisäosa, jolla voidaan renderöidä 3D-mallista epärealistisia, tyylieltyjä viivapiirroksia. (Kuva 1) Freestyle on suunniteltu antamaan käyttäjälleen maksimaalisen kontrollin lopputulokseen: Käyttäjälle tarjotaan paljon asetuksia joiden avulla määritetään miten siluetti ja muut 3D-mallin piirteet muutetaan tyylieltyiksi viivoiksi. (Kuva 2) (Sourceforge Freestyle 2005)

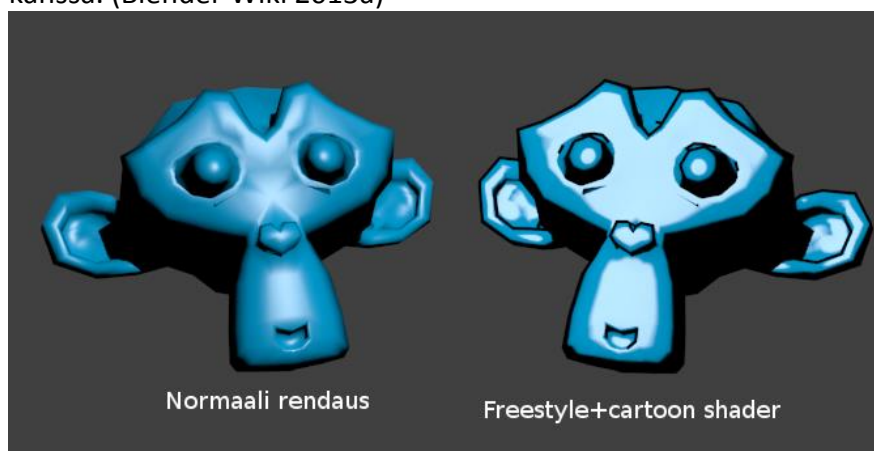


Kuva 1. Eri tyyliä ääri viivoja freestyle-moottorilla renderöitynä



Kuva 2. Viivojen määrittelyyn käytettävät asetukset

On huomioitavaa että Freestyle luo vain viivat eli kappaleen pintojen värit ja varjot jää muiden render-moottorien laskettavaksi. (Kuva 3) Freestyle on kuitenkin ingetroitu render layereihin ja kompositoriin, joten freestyle-viivapiirroksia on helppo yhdistää muiden renderöityjen komponenttien kanssa. (Blender Wiki 2013a)

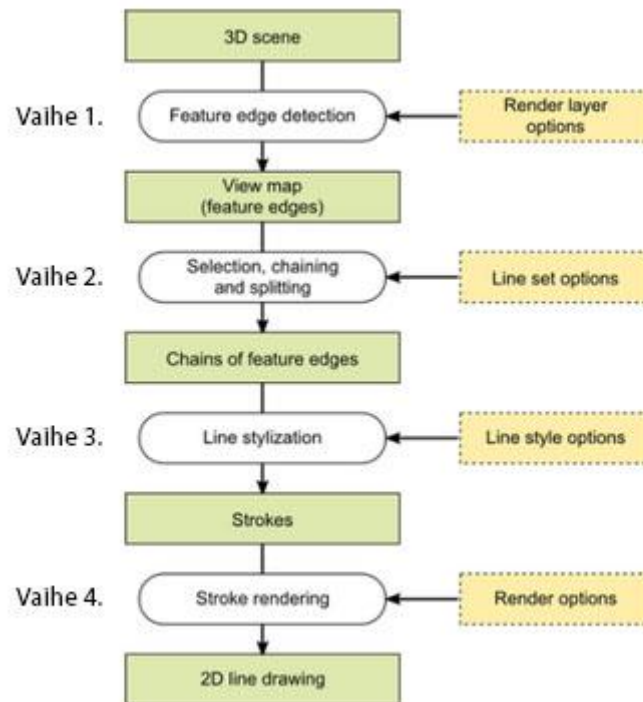


Kuva 3. Normaali renderöinti vs. Freestylellä tehdyt viivat yhdistettynä cartoon shaderiin.

2.1.1 Freestylen renderöintiprosessi

Freestyle luo 2D-viivapiirroksen mesh-objekteista. Meshin vertexejä, edgejä ja faceja käytetään objektin piirteiden tunnistamiseen. Tunnistettujen piirteiden perusteella piirteet ja yksityiskohdat luodaan tyyllitellyiksi viivoiksi.

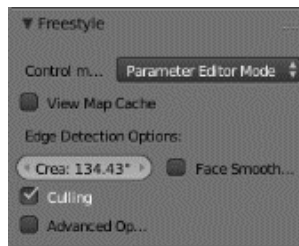
Freestyle muuttaa 3D-kohtauksen 2D-viivapiirrokseksi seuraavalla tavalla (kuva 4):



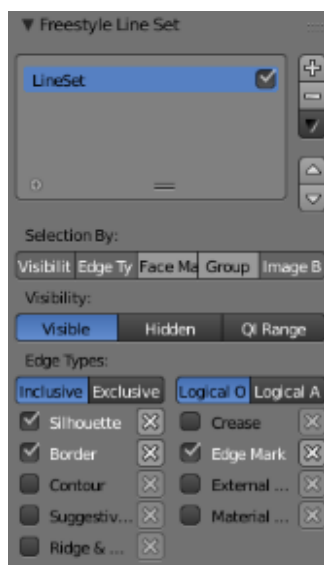
Kuva 4. Diagrammi freestylen render-prosessin eri vaiheista. Keltaisissa laatikoissa nimetään asetukset, millä vaikutetaan missäkin prosessin vaiheessa.

1. Feature edge detection. Ensimmäisessä vaiheessa freestyle tunnistaa 3D-objektista piirteitä käyttäjän render layers-sivulla määrittämien asetusten mukaan. (Kuva 5) Piirteitä tunnistava algoritmi tunnistaa tärkeät yksityiskohdat, kuten siluetit, rajat ja rypyt. Tunnistetuista yksityiskohdista luodaan ohjelman välimuistiin viivajoukko, tätä viivajoukkoa kutsutaan nimellä view map. View map luodaan erikseen jokaiselle render layerille. View map-vaiheessa viivoja käsitellään vektoreina perinteisen bittikarttagrafiikan sijasta. Tämä antaa käyttäjälle maksimaalisen kontrollin lopputulokseen.
2. Selection, chaining and splitting. Toisessa vaiheessa moottori valitsee manipuloitavaksi view mapin viivoja käyttäjän asettamien asetusten perusteella. (Kuva 5) Freestyle yhdistää ja ketjuttaa valituista viivoista pohjan, jota käytetään prosessin seuraavassa vaiheessa. Valittua

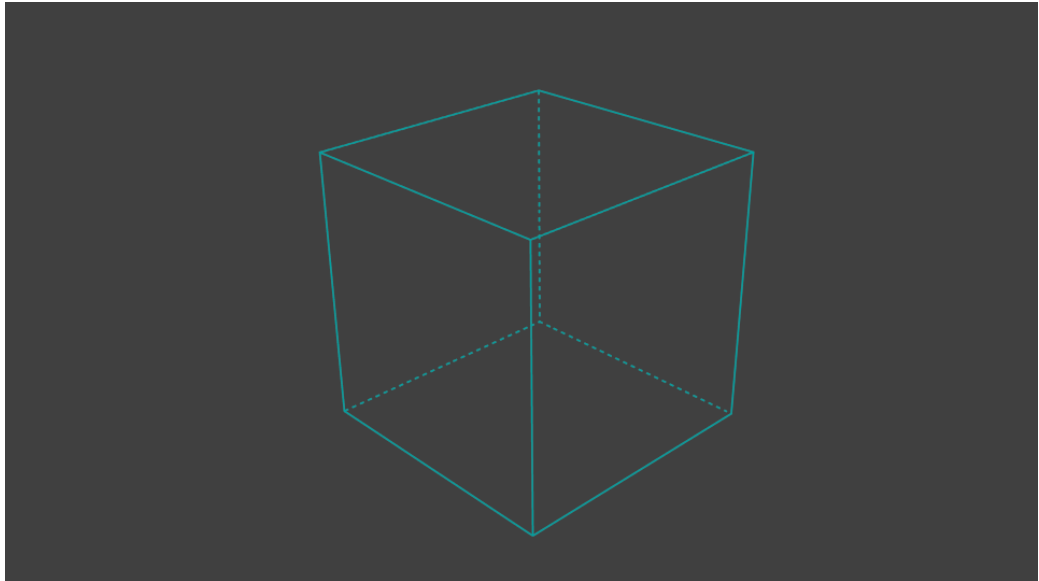
viivajoukkoa kutsutaan LineSetiksi. Yhden view mapin pohjalta voi luoda useita eri LineSettejä. Tämä mahdollistaa esimerkiksi näkyvien sekä piilossa olevien viivojen samanaikaisen renderöinnin. (Kuva 6 ja Kuva7)



Kuva 5. Render Layers-asetukset freestyle-moottorille

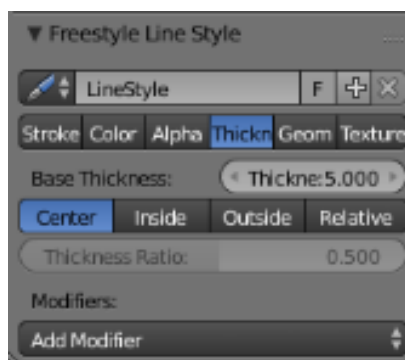


Kuva 6. LineSet-asetukset



Kuva 7. Yhdellä renderöinnillä tehty kuva kuutiosta. Renderöinnissä käytetty yhtä view mappia, josta on määritetty kahta eri LineSettiä (asetuksilla visible ja hidden) joille molemmille määrittelin eri LineStyle-asetukset (yhtenäinen viiva ja katkoviiva).

3. Line stylization. Kolmannessa vaiheessa valmis LineSet muutetaan tyyllitellyiksi ääri viivoiksi käyttäjän määrittämien tyyli valintojen mukaisesti. Tyyli valintoja kutsutaan yhteisesti nimellä line style. (Kuva 8) Freestyle-moottori takaa käyttäjälle suuren valikoiman asetuksia joilla viivoista voi tehdä juuri sen näköiset kuin käyttäjä itse haluaa. Tallennettuja Line Style-asetuksia voi käyttää myös muissa Line Seteissä ja jopa muissa Blender-tiedostoissa.



Kuva 8. Line style-asetukset

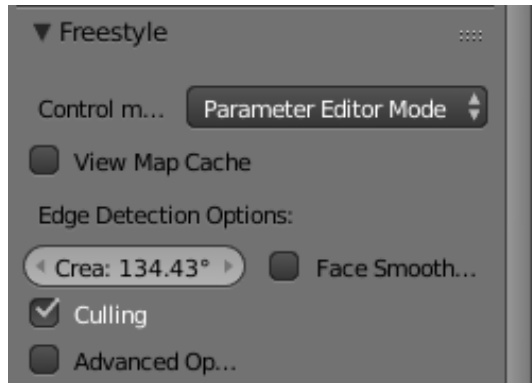
Stroke rendering. Neljännessä vaiheessa tyyllitellyt viivat renderöidään kuvaksi ja lopputuloksena saatavat viivat yhdistetään Blenderin sisäisen render-moottorin laskemaan kuvaan. (Blender Wiki 2013)

4. Stroke rendering. Neljännessä vaiheessa tyyllitellyt viivat renderöidään kuvaksi ja lopputuloksena saatavat viivat yhdistetään Blenderin sisäisen render-moottorin laskemaan kuvaan. (Blender Wiki 2013)

2.2 Freestylen asetukset

2.2.1 Viivojen tunnistaminen objektista

Jokaiselle render layerille tulee erikseen määrittää omat asetukset viivojen tunnistamiseen. (Kuva 9)

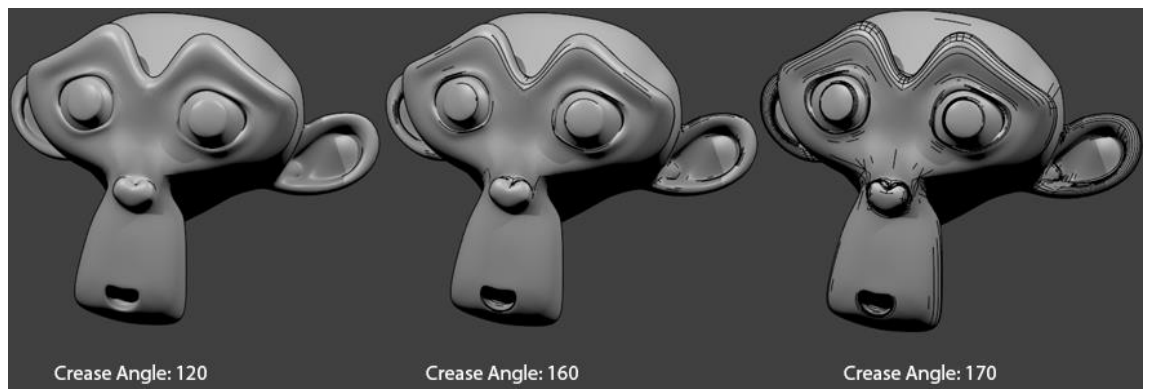


Kuva 9. Freestyle-asetukset render layereille

Crease angle: Kontrolloi uurteiden ja ryppyjen näyttämistä (mitä isompi kulma, sitä enemmän viivoja) (Kuva10)

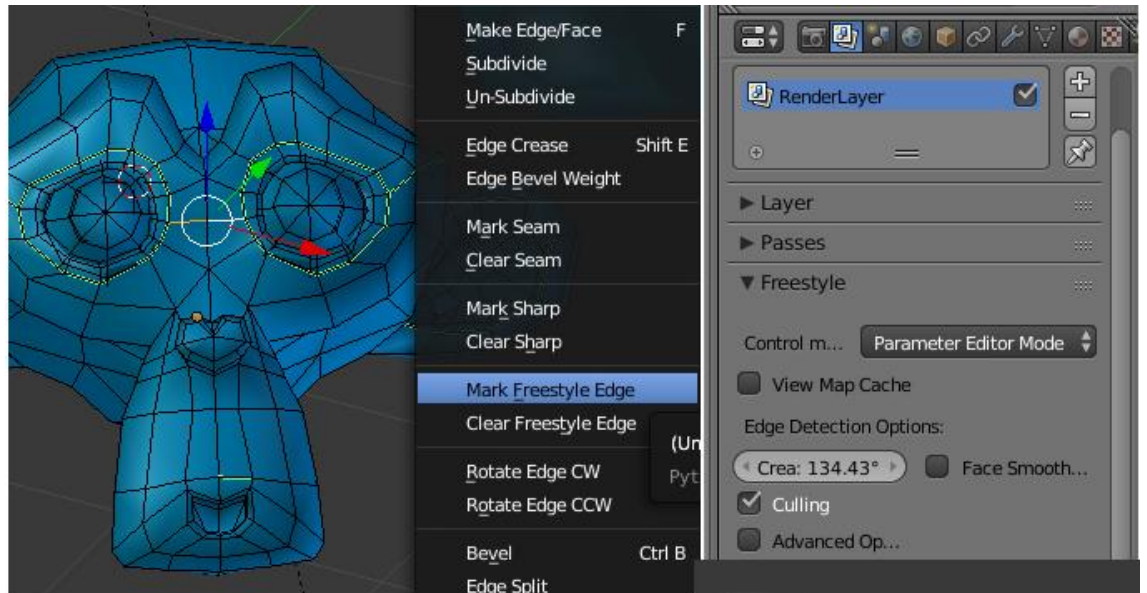
Culling: Jättää huomiotta viivat jotka ovat 2D-kuvan rajojen ulkopuolella (lähinnä suorituskykyä parantava vaikutus)

Face smoothness: Huomioi kappaleen pehmeän muodon jolloin siluetin viiva on pehmeämpi (pidempi kesto-renderointi). (Blender Wiki 2013)

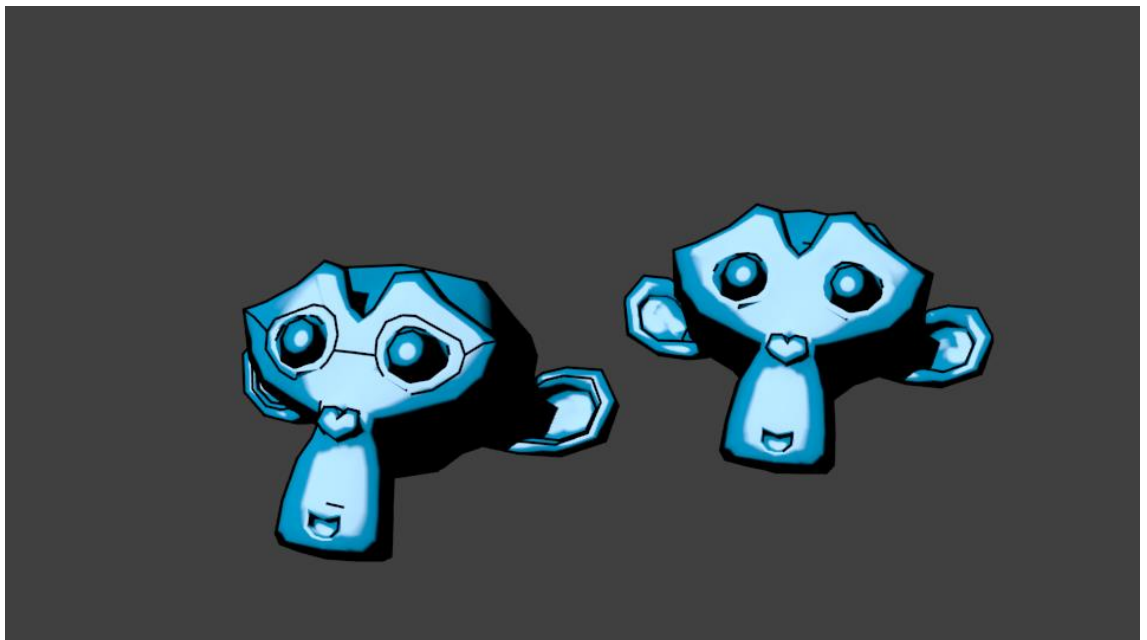


Kuva 10. Crease Anglen vaikutus viivojen tunnistamiseen

Käyttäjä voi myös mallintaessaan merkata edgejä tai faceja käyttämällä Freestyle Edge Mark-ominaisuutta. Merkatut edget voi joko pakottaa näkyviin tai piilottaa niin, että niitä ei piirretä ollenkaan. (Kuva 11 ja Kuva 12) Merkatut viivat asetetaan näkyviin LineSet-asetuksista. (Blender Wiki 2013)



Kuva 11. Freestyle edgen merkkäminen sekä render layerin viivojen määrittelyasetukset. Kuvassa näkyvät vihreät edget on merkattu Freestyle edgeiksi.



Kuva 12. Edge Markeilla pakotetut "silmäläsit" ja toinen apina ilman merkattuja edgejä

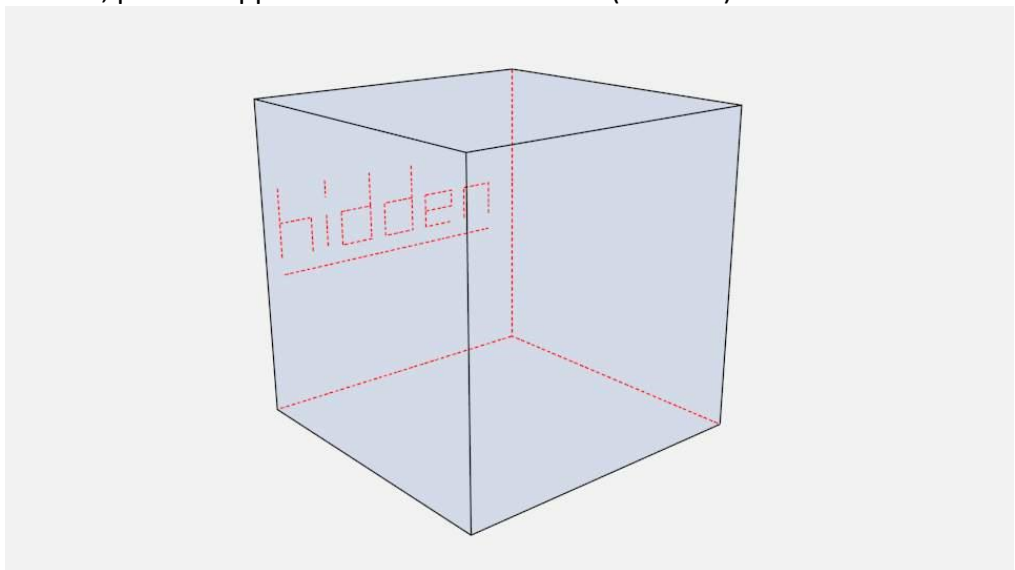
2.2.2 LineSet

LineSet valitsee freestylen ensimmäisessä vaiheessa luoman view mapin viivoista käyttäjän valitseminen asetusten perusteella, mitkä viivat tullaan renderöimään. Viivojen valintaan on kolmen eri perustyyppin työkalua: näkyvyys, edge-tyyppi ja käyttänä itse pakottamat valinnat. Näiden perustyyppien alta löytyy useita eri valintoja:

Selection by visibility – sisältää kolme eri tapaa näkyvyyden mukaan valitsemiseksi:

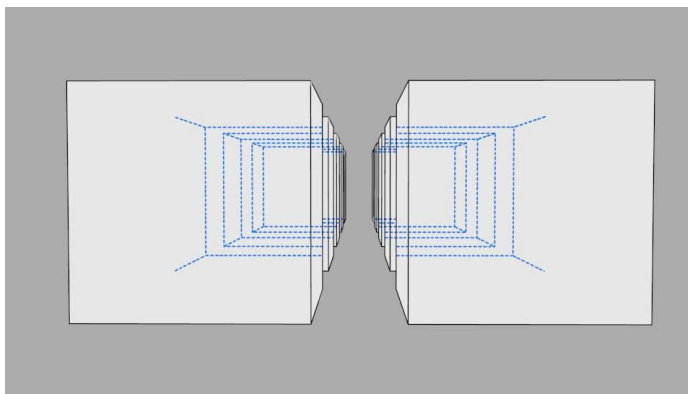
Visible, piirtää näkyvissä olevat viivat (Kuva 13)

Hidden, piirtää kappaleen takana olevat viivat (Kuva 13)



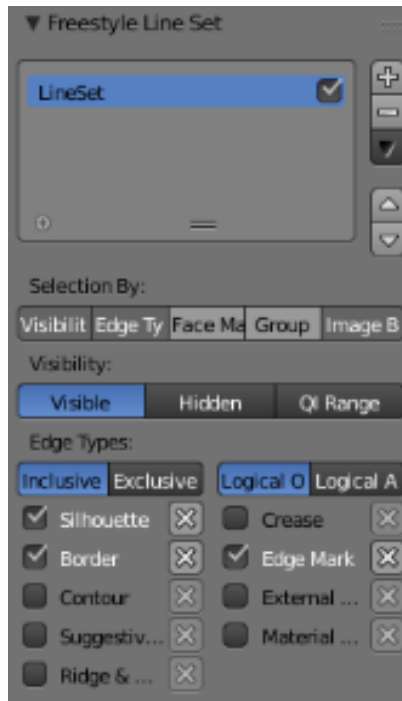
Kuva 13. Punaisella näkyy kappaleen takana olevat viivat. Kuvassa myös edgemarkkeilla pakotettu hidden-teksti.

QI Range eli Quantitive Invisibility, käyttäjä määrittää kuinka monen pinnan takana olevat viivat valitaan piirrettäviksi (tässä metodissa pitää määrittää end ja start arvot.) (Kuva 14)



Kuva 14. Kuvassa on määritetty QI Range arvoille 3-7.

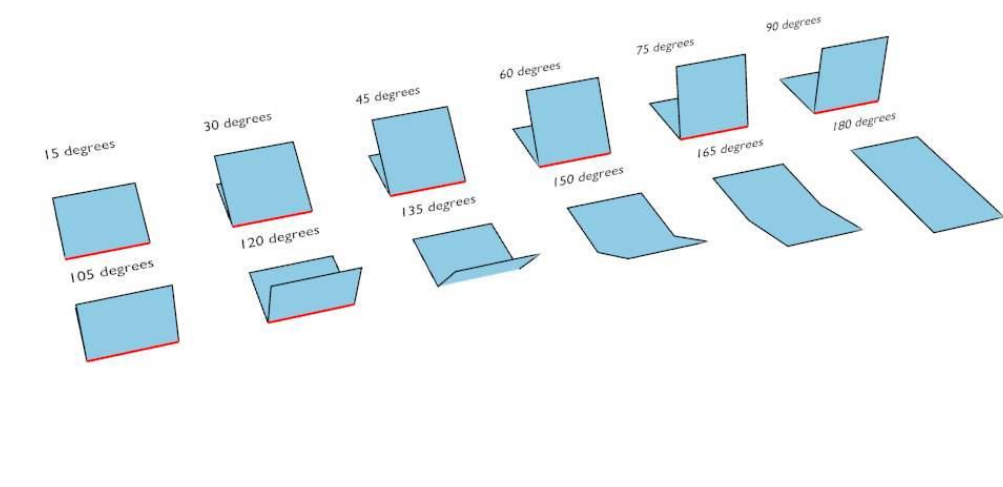
Selection by edge types – edge-tyyppien valintaan käytetään perus algoritmeja geometrian erilaisten viivojen valitsemiseksi. Käyttäjän täytyy tehdä vähintään yksi valinta, jotta moottori voi renderöidä kuvan, mutta useampien tyyppien yhdistely on mahdollista saman linesetin sisällä. (Kuva 15) Edge tyyppejä voi myös tiputtaa pois laskenta-prosessista painamalla X:ää valinnan vieressä, tämä nopeuttaa renderöintiä.



Kuva 15. LineSet-asetukset.

Crease:

Näyttää vain edget joiden viereiset facet muodastavat isomman kulman, kuin käyttäjän viewmapille määrittämä crease angle. (Kuva 16)



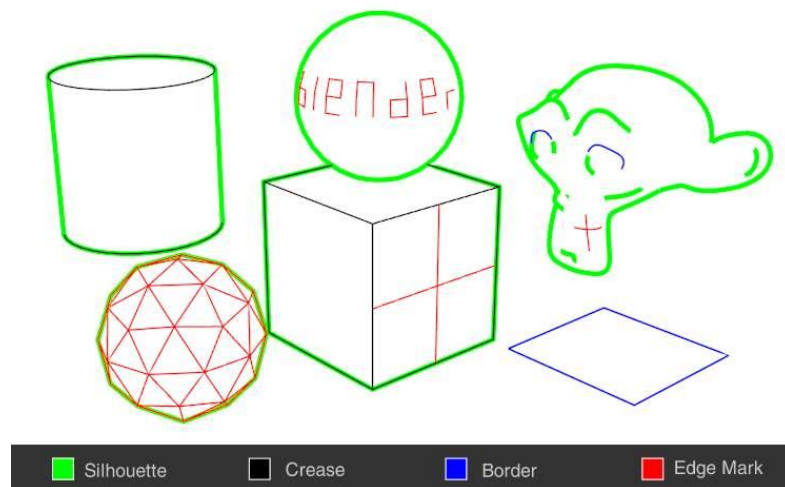
Kuva 16. 121-asteen kulmalla tehty havainnollistus Crease-valinnalle

Silhouette:

Piirtää siluetin objektin ympärille. Toimii hyvin orgaanisilla objekteilla, kuten apinalla, ja pallolla, mutta huono objekteilla joissa on teräviä reunoja esim. kuutio. (Kuva 17) Silhouette ei ota huomioon avoimien objektien reunoja, kuten sylintereitä tai tasaisia tasoja (plane.)

Border

Border on tarkoitettu avoimien meshien (edget joiden välissä ei ole facea eli pintaa) valitsemiseen; avoimella sylinterillä on yläreunassa avoin edge, plane on avoin kokonaan avoin sekä apinan silmäkuoppa on avoin. Kaikki avoimet edget piirretään. (Kuva 17)



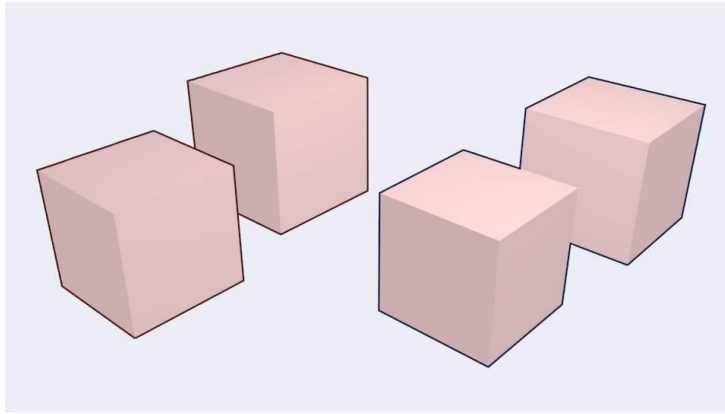
Kuva 17. esimerkkejä erilaisista edge-valinnoista. Oikeassa alakulmassa neliö johon vaikuttaa vain border valinta, koska mesh on joka sivulta avoin.

Contour

Piirtää objektin ääri viivat (Kuva 18)

External Contour

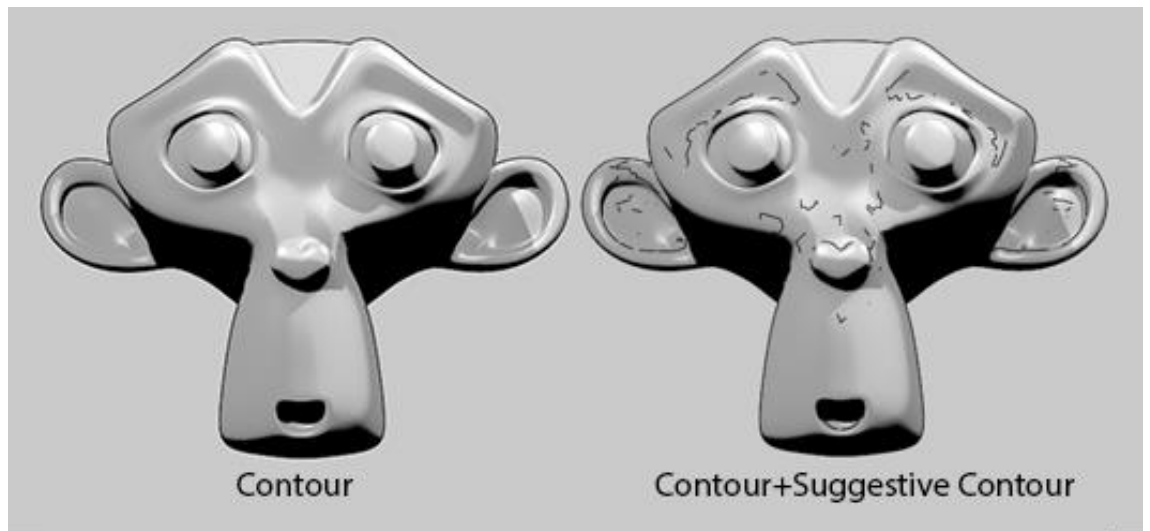
Piirtää vain uloimman ääri viivat (Kuva 18)



Kuva 18. Vasemmalla contour- ja oikealla external contour-valinta.

Suggestive Contour

Piirtää viivoja jotka muodostavat ääri viivoja toisesta kuvakulmasta katsottaessa. Suggestive contour täydentää yksityiskohtia, jotka jäisivät piirtämättä pelkällä contour valinnalla. (Kuva 19)



Kuva 19. Suggestive contour-valinnalla piirretyt viivat tuovat esiin enemmän yksityiskohtia, kuin pelkällä contourilla

Material boundary

Piirtää viivan kohtaan jossa kaksia eri materiaalia kohtaavat.(Kuva 20)
(Blender Manual n.d.a)

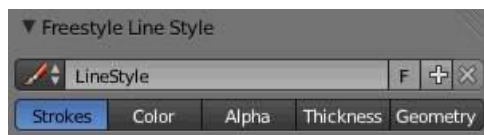


Kuva 20. Maskin rajaava musta ääriiviiva piirtyy material boundary-valinnalla

2.2.3 LineStyle ja modifierit

Freestyle-renderöinnissä "line style" määrittää viivojen ulkoasun viiden eri ominaisuuden mukaan: (Kuva 21)

- Stroke
- Color
- Alpha
- Thickness
- Geometry



Kuva 21. Line style-asetuksien valikko

Nämä asetukset mahdollistavat viivojen renderöinnin eri tyyleillä ja eri käyttötarkoituksiin (tekninen piirustus, karkea luonnostelma, sarjakuva, kalligrafia jne.) (Kuva 22)

Voit luoda niin monta line stylea kuin haluat ja uudelleen käyttää tyyliä eri line seteissä valitsemalla haluamasi tyylin pudotusvalikosta line setin nimen vierestä.

Huom. Jos ei toisin määritellä niin kaikki mitat line style-asetuksissa on pikseleinä (joko relatiivisina tai absoluuttisina). (Blender Manual n.d.b)

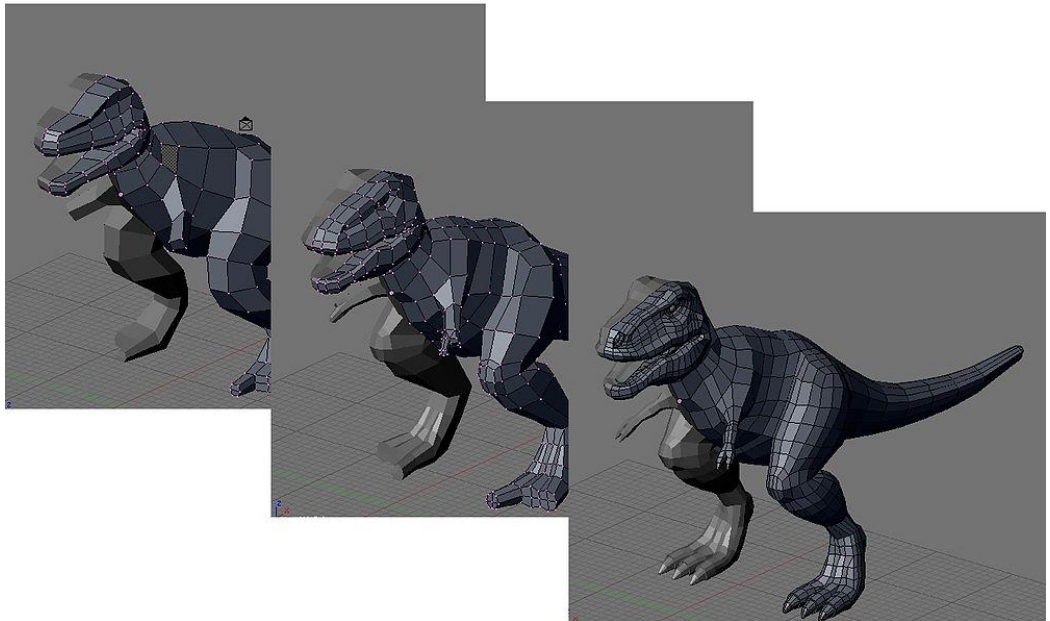


Kuva 22. Esimerkkejä eri asetuksilla renderöidyistä viivoista (Blender Manual n.d.)

2.3 Mallintaminen

2.3.1 Box modeling

Box modeling tarkoittaa 3D-mallinnus tekniikkaa, missä artisti aloittaa matala resoluutioisesta kappaleesta (yleensä kuutio tai pallo) ja muokkaa kappaleen muotoa pursottamalla, skaalaamalla tai kääntelemällä pintoja ja edgejä. Yksityiskohtia lisätään käsin joko lisäämällä edge looppeja tai jakamalla kappale pienemmiksi pinnoiksi. (Kuva 23) (About tech n.d.a)

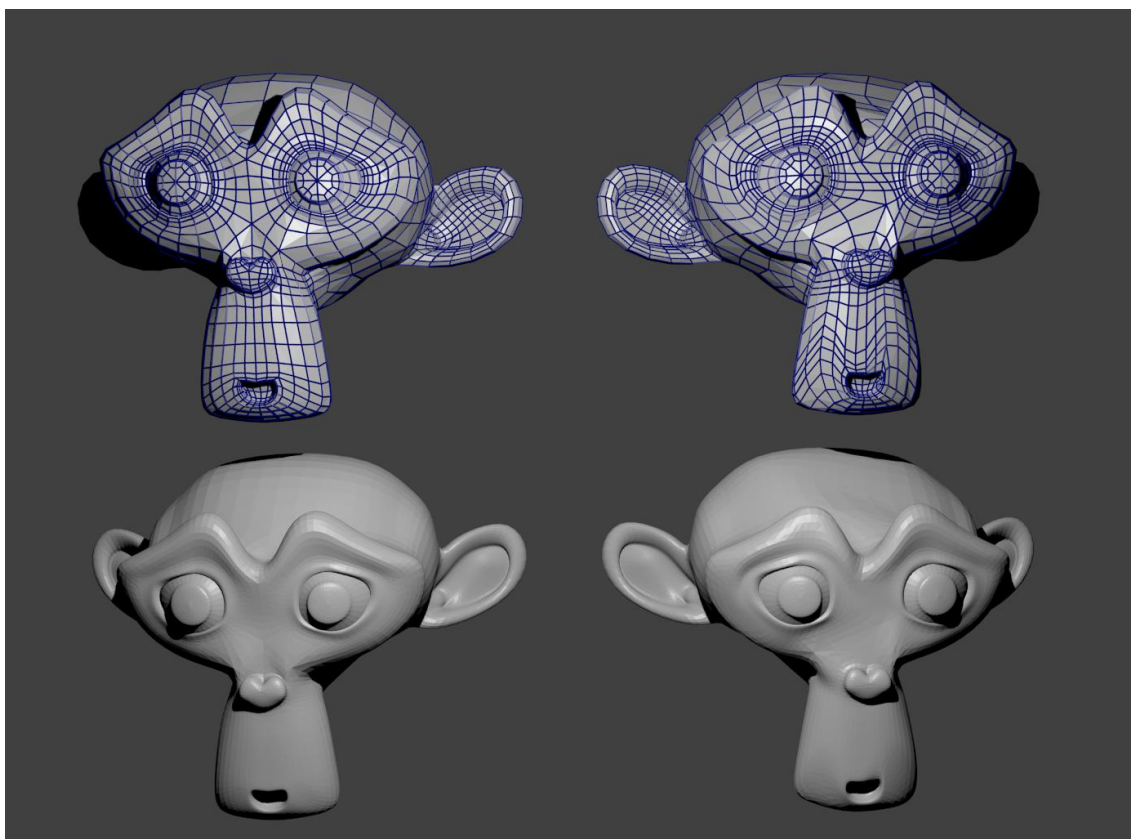


Kuva 23. yksityiskohtien lisääminen kappaleeseen box modeling-tekniikalla. Huomaa miten polygonien määrä lisääntyy yksityiskohtia lisätessä. (Blender Wiki 2016a)

2.3.2 Topologia

Sanalla topologia viitataan 3D-kappaleen pinnan geometrisiin ominaisuuksiin. Mallintajat tähtäävät ”puhtaaseen” topologiaan, mikä yleensä esitetään 3D-rautalankamallina, jossa on suorituskyvyn kannalta järkevä määrä polygoneja, jotka on yksityiskohtia silmällä pitäen jaettu järkevästi kappaleen pinnalle ja edge loopit on sijoitettu oikeisiin kohtiin. (Kuva 24)

Hyvää topologiaa rakentaessa on otettava huomioon myös kappaleen animoitavuus. Alueet jotka muuttuvat eniten animoitaessa vaativat suuremman määrän polygoneja (nivelet, kasvot jne.) (About tech n.d.b)



Kuva 24. Vasemmalla hyvä topologia ja oikealla huono topologia. Alempia malleja verrattaessa on vaikea nähdä eroja, mutta ylemmät rautalanka-mallit paljastavat kappaleiden erot. Oikean puoleisen kappaleen topologia on vääntynyt ja edget mutkittelevat erityisesti suun ja nenän yläpuolella.

2.4 Rig ja animointi

Riggaamisella tarkoitetaan meshiä liikuttavaa luurankoa. Koska kappaleen jokaisen pisteen erikseen animoiminen olisi erittäin aikaa vievää, luomme luurangon jonka luita mesh seuraa. (Kuva 25)

Blenderissä tätä luurankoa kutsutaan nimellä armature. Armature toimii seuraavalla tavalla:

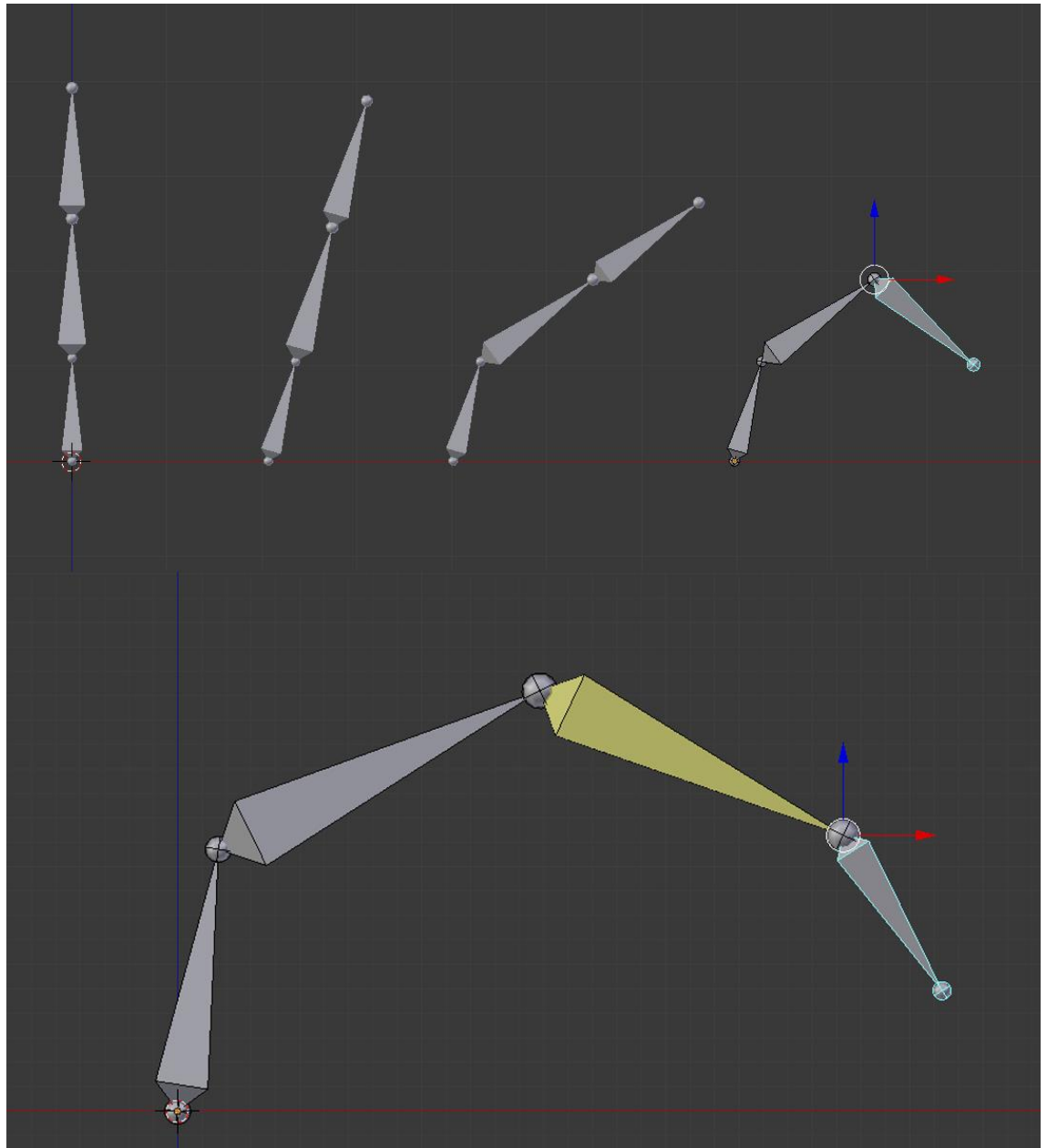
- Armature koostuu useasta luusta jotka muodostavat ketjuja (esim. yksi raaja on yksi ketju).
- Jokainen luu on yhdistetty kappaleen vertex-pisteisiin.
- Kun liikutat luuta vain siihen yhdistetyt pisteet liikkuvat, muut pisteet pysyvät paikoillaan.
- Tavoitteena on asettaa luut niin, että vertexit liikkuvat pehmeästi ja luonnollisesti. (Blender Wiki n.d.b)



Kuva 25. Rig ja sen vaikutus meshiin.

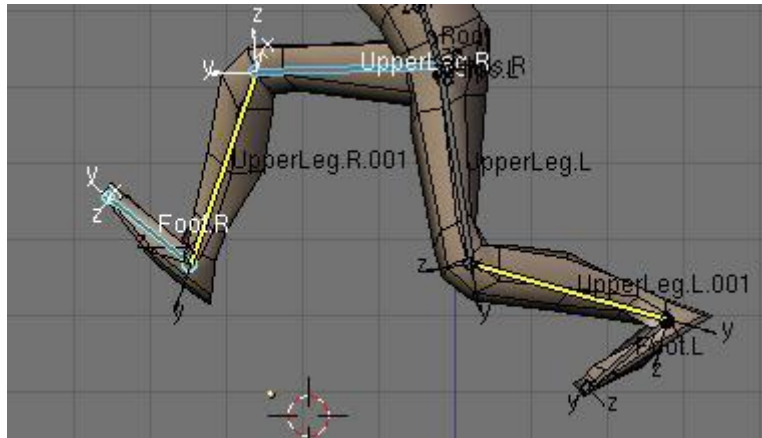
2.4.1 Inverse kinematics

Inverse kinematics eli IK on forward kinematicsin vastakohta. IK yksinkertaistaa animointi prosessia ja mahdollistaa monimutkaisten animaatioiden tekemisen pienemmällä vaivannäöllä. IK antaa mahdollisuuden määrittää luuketjulle kontrolliluun, jolloin muut ketjun luut seuraavat sitä. Forward kinematics tekniikalla animoijan pitää aloittaa ketjun ensimmäisestä luusta ja liikuttaa luita järjestyksessä, kunnes ketjun kaikki luut on liikutettu haluttuun asentoon: Kun luuta liikuttaa hierarkiassa sen alapuolella olevat luut perivät luun sijainnin ja pyörytyskulman. Tämä tekee pienien muutoksien tekemisestä vaikeampaa, koska animoija joutuu mahdollisesti liikuttamaan kaikkia luita saadakseen halutun lopputuloksen. Tämä ongelma voidaan pitkälti kiittää käyttämällä IK-menetelmää. (Kuva 26) (Blender Manual n.d.c)



Kuva 26. Ylempänä FK-ketju josta jouduin liikuttamaan kaikkia luita. Lähdin liikkeelle ketjun ensimmäisestä luusta ja käänsin luita kunnes kaikki luut oli halutussa asennossa. Alempana IK-ketju josta olen liikuttanut pelkästään yhtä luuta, mutta muut luut seuraavat mukana koska, ketjun molemmat päät ovat luiden kohteina. Huomaa että IK-ketjussa on yksi ylimääräinen luu joka kontrolloi ketjun muita luita.

Hyvä esimerkki inverse kinematicsin käyttämisestä on jalka. Jalka on kiinni vartalossa ja loppuu jalkaterään, määrittämällä jalkaterän luun kontrolliluuksi voi koko jalan asennon muuttaa liikuttamalla pelkästään jalkaterän luuta. (Blender Wiki n.d.d) (Kuva 27)

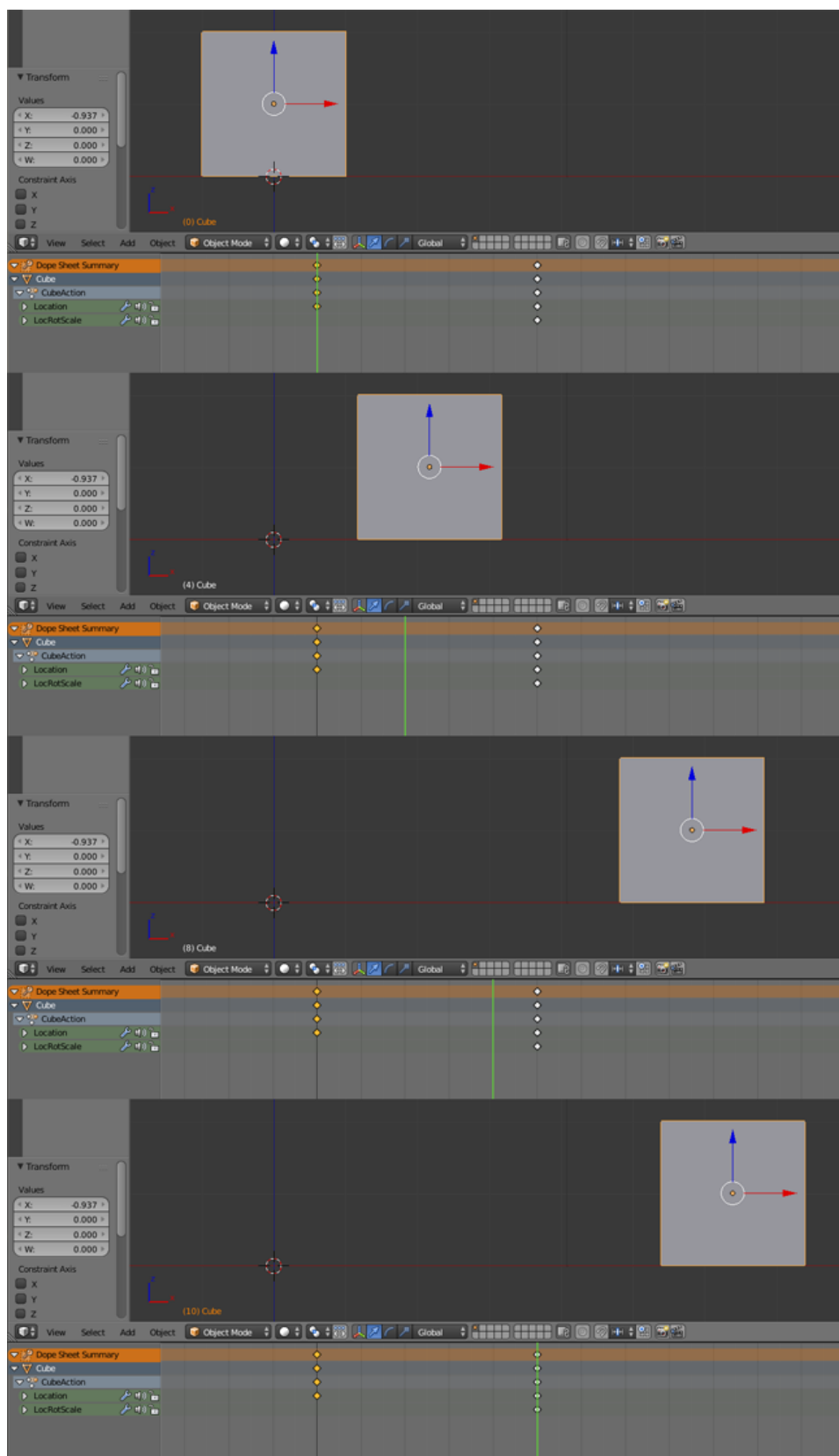


Kuva 27. Inverse Kinematics jalkojen liikuttelussa. Tässä tilanteessa koko jalka liikkuu jos pelkästään jalkaterän luuta liikutetaan.

2.4.2 Keyframe-animointi

Keyframe on merkki aikajanalla johon tallennetaan arvo kyseisellä ajan hetkellä. Esimerkiksi key frame voi osoittaa, että 0. frameissa objektin sijainti X-akselilla on 0.

Keyframen tarkoitus on mahdollistaa animointi pisteiden välillä. Esimerkiksi käyttäjä voi lisätä toisen key framen 10. frameen ja määrittää objektin sijainniksi 6. Tämän jälkeen Blender pystyy määrittämään objektin sijainnin kaikille frameille näiden kahden pisteen väliin, jolloin objekti liikkuu sulavasti määritettyjen pisteiden välin. (Blender Manual n.d.d) (kuva 28)



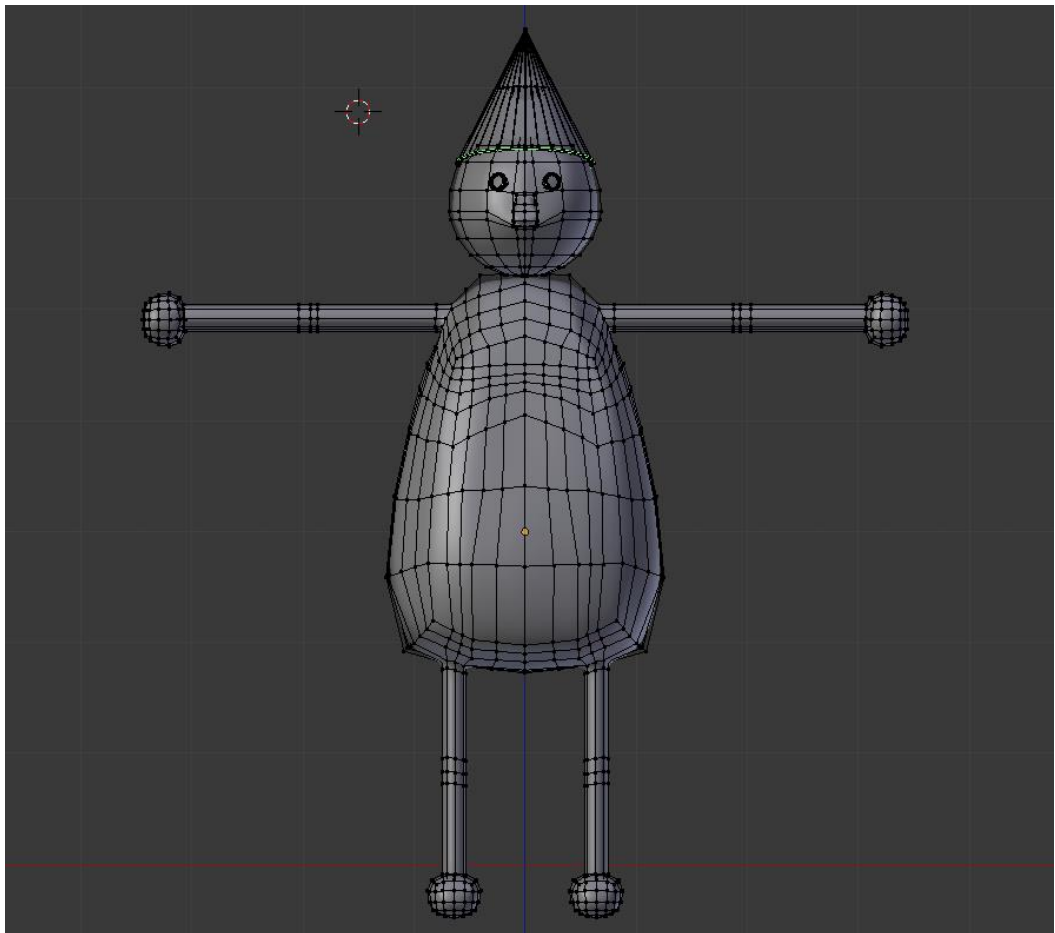
Kuva 28. Kuvassa keyframe-animointi yksinkertaisimmillaan. Kuution sijainti X-akselilla 0 framessa on nolla ja 10. framessa 6. Huomaa että vaikka aikajanalle on määritetty vain kaksi pistettä, frameen 0 ja 10. frameen, silti kuutio liikkuu jokaisella framella. Vihreä viiva kuvaa aikajanana lukupäätä.

3 2D-ANIMAATION TYÖNKULKU

3.1 Mallinnus

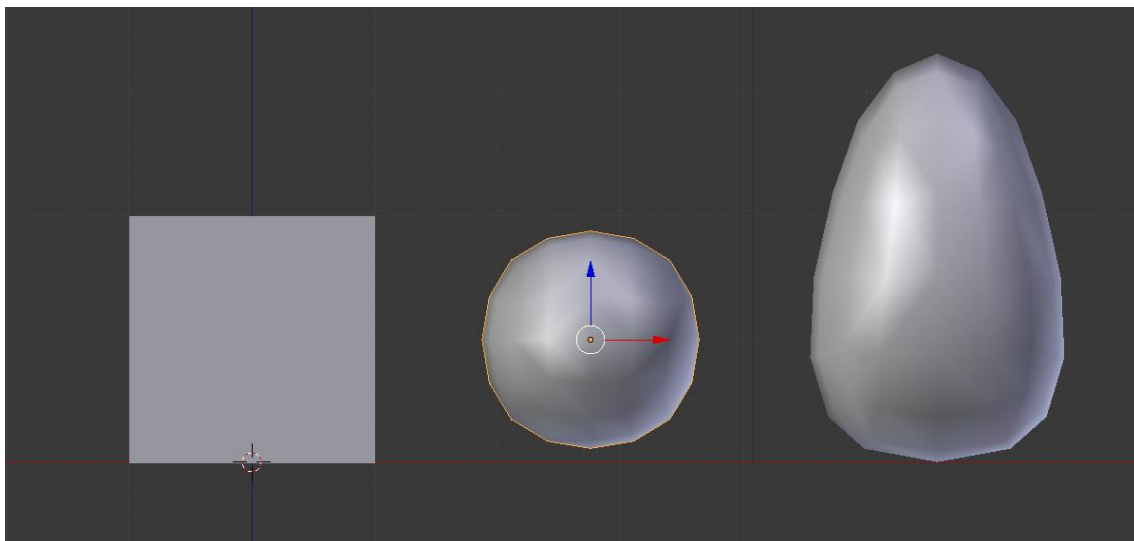
3.1.1 Box modeling

Tässä työssä mallintaminen oli todennäköisesti projektin helpoin osuus. Referenssinä käytin YouTube-videota ruotsin esityksestä, josta karkeasti kopioin hahmon ulkoasun. Hahmon yksinkertaisen ulkomuodon ansiosta varsinainen mallinnus painottui hahmon nenän ja vartalon aikaansaamiseksi, loppu hahmosta koostuu muutamasta eri primitiivistä, jotka asettelin raajoiksi ja muiksi osiksi. Animoiminen huomioon ottaen mallinoin hahmon perinteiseen T-asentoon, jonka tarkoitus on helpottaa riggaamista. Lisäksi lisäsin raajoihin edge loopit joiden tarkoitus on vahvistaa niveliä animointia varten. (Kuva 29)



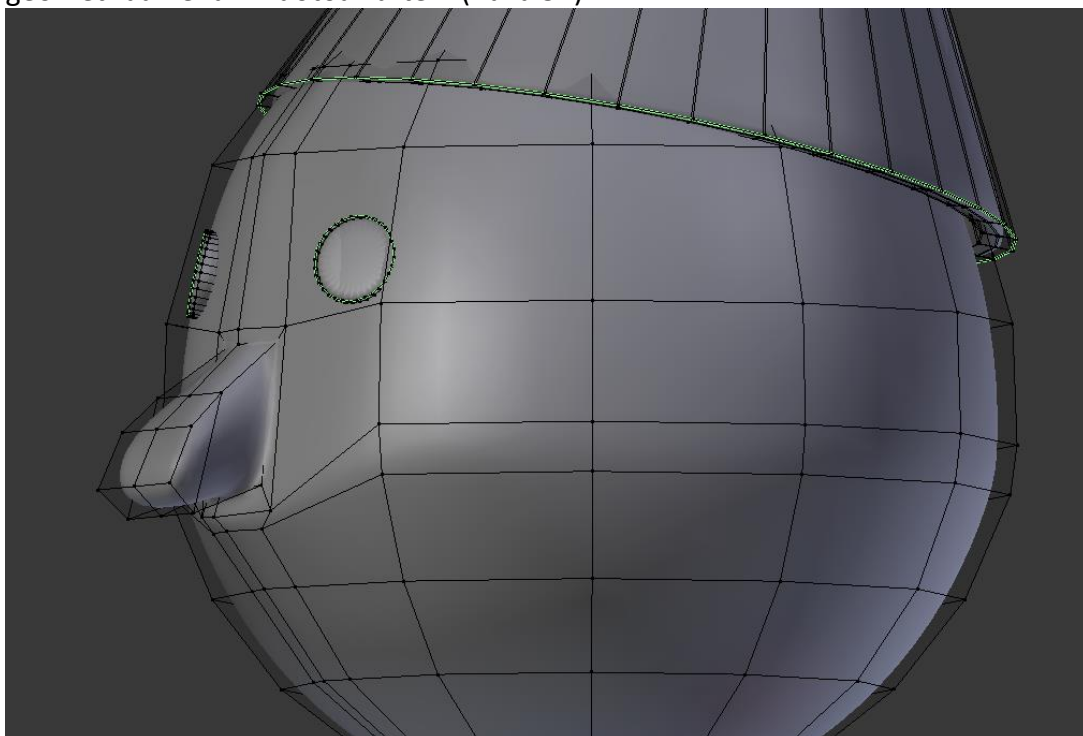
Kuva 29. Päätä ja vartaloa lukuunottamatta rakensin hahmon yksinkertaisista primitiiveistä. Rendaustavan takia topologialla ei ollut tässä työssä suurta merkitystä.

Vartalon mallinsin niin, että lisäsin kuution, jonka jaoin kaksi kertaa subdivision-modifierilla, saadakseni kappeleeseen tarpeeksi suuren resoluution oikean muodon aikaansaamiseksi. Tätä kappaletta venyttämällä tein vartalon lopullisen muodon. (Kuva 30)



Kuva 30. Vartalon kehittäminen primitiivistä. Ensimmäisenä kuutio, minkä jaoin subdivision-modifierilla, tämän operaation jälkeen kuution muoto muistuttaa enemmän palloa, kuin kuutiota.

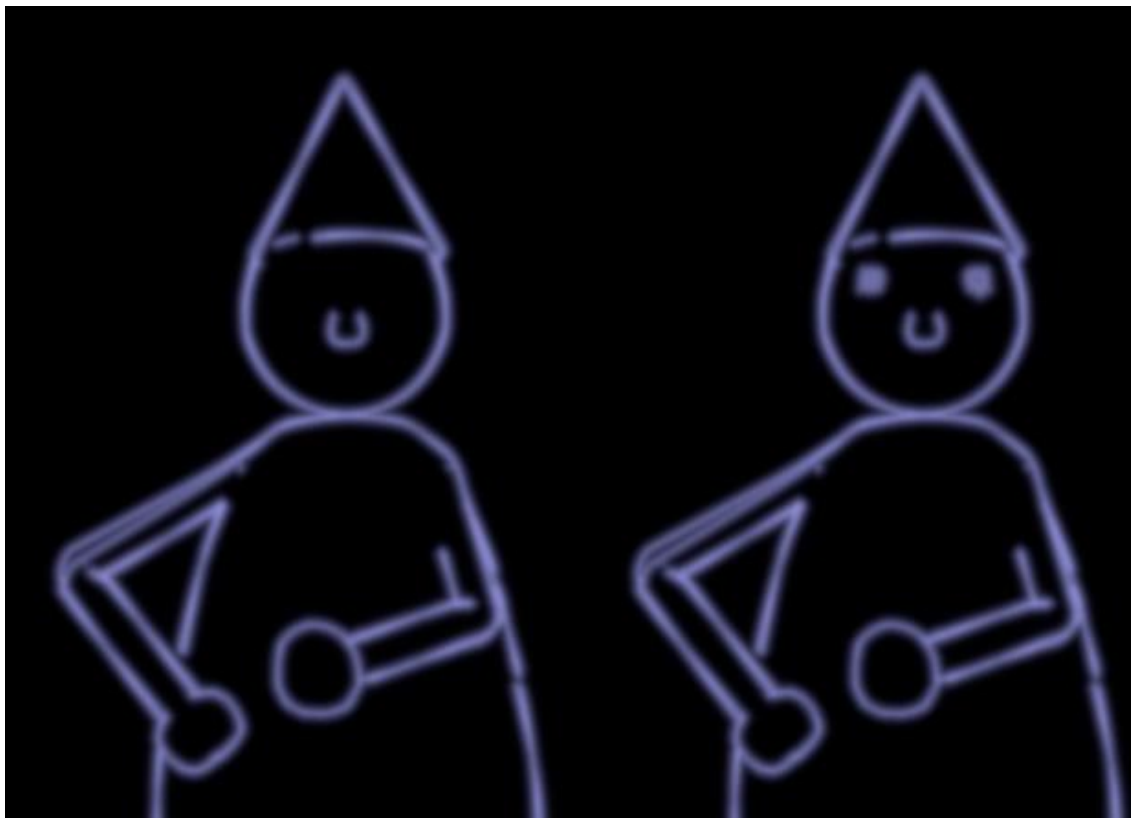
Nenä on yksinkertaisella pursotus-tekniikalla tehty uloke pallossa. Jouduin jakamaan nenän kohdan kahdella edge loopilla saadakseni lisää geometriaa nenän muotoa varten. (Kuva 31)



Kuva 31. Pää on pallo johon jaoin parilla edgeloopilla lisää geometriaa nenää varten.

3.1.2 Face Markien määrittely

Pakotin silmien ja hatun reunan viivat näkyviin facemarkien avulla, koska hahmon liikkessa oli tilanteita, joissa freestylen viivojen tunnistus algoritmi ei tunnistanut näitä viivoja riittävän hyvin. (Kuva 32)

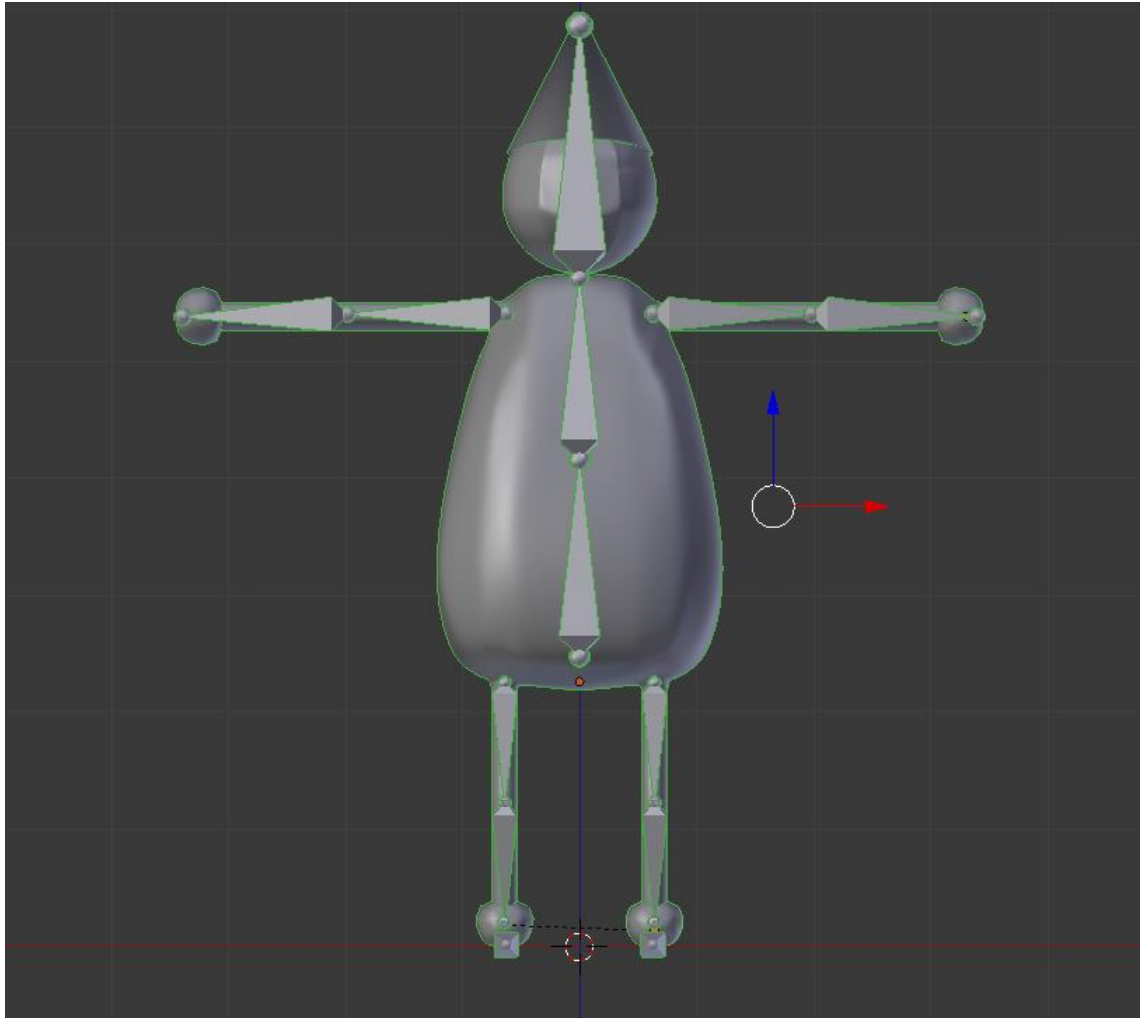


Kuva 32. Silmät piti pakottaa näkyviin facemark-toiminnolla. Vasemmalla rendöity kuva ilman facemarkeja ja oikealla niiden kanssa.

3.2 Riggaaminen

3.2.1 Rig

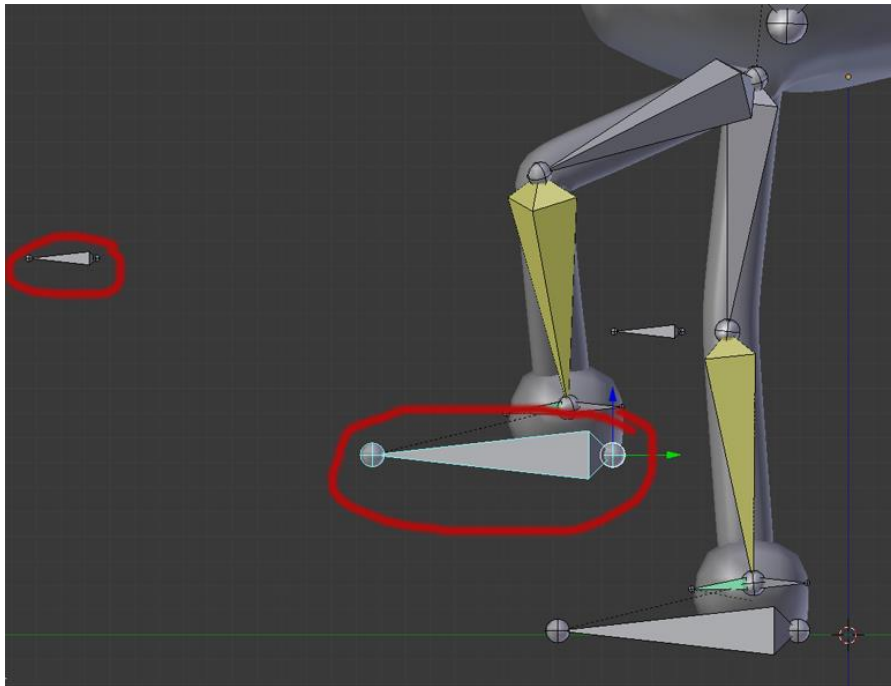
Seuraava työvaihe mallintamisen jälkeen on riggaaminen. Koska hahmolla oli tarkoitus tehdä vain hyvin yksinkertaisia liikkeitä ja animaatioita, ei rigin tarvinnut olla monimutkainen. Tein rigiin vain tarvittavat luut: vartalolle, päälle ja raajoille. Aloitin rigin tekemisen lisäämällä vain keskilinjan ja vasemman puolen luut. Tämän jälkeen nimesin luut seuraavalla tavalla: "LuunNimi.L", ".L" osoittaa, että luu on vasemmalla puolella hahmoa. Näin tehtäessä Blender ymmärtää, että on kyse vasemman puolen luista. Tämän jälkeen kopioin vasemman puolen luut, peilasin ne hahmon oikealle puolelle ja käytin automaattista nimeämistoimintoa joka muuttaa peilattujen luiden nimet muotoon: "LuunNimi.R". Kun kaikki luut oli asetettu paikoilleen yhdistin meshin ja rigin tekemällä rigistä meshin parentin. (Kuva 33)



Kuva 33. Kuvassa hahmo jonka päällä hahmon rig eli luuranko.

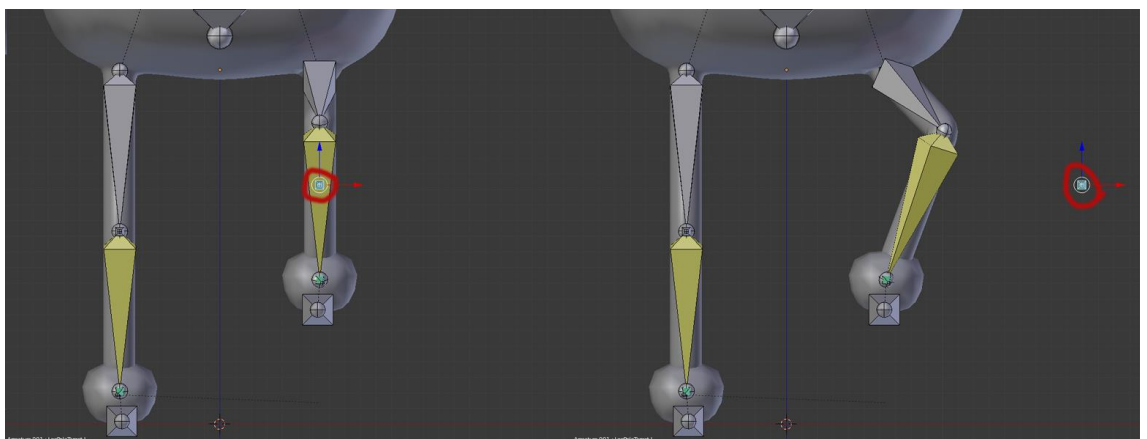
3.2.2 Inverse Kinematics

Käytin IK-menetelmää jalkoihin, jalkojen liikuttelun helpottamiseksi. Ohje jota käytin riggaamisen apuna neuvoi jostain syystä, että käsien animoiminen on helpompaa ilman IK-menetelmää. Tottelin ohjetta ja kaduin tätä animoitaessa niin paljon, että yritin lisätä IK-kontrollin käsiin jälkikäteen, mutta se ei ollut enää mahdollista. (Kuva 34)



Kuva 34. Ympyröitynä jalan kontrolliluu (jalkaterä) ja pole target (leijuu ilmassa) joka määrittää, mihin suuntaan raaja osoittaa. Samalla havainnollistan miten yhtä luuta liikuttamalla on koko raajan asento kokonaan muuttunut.

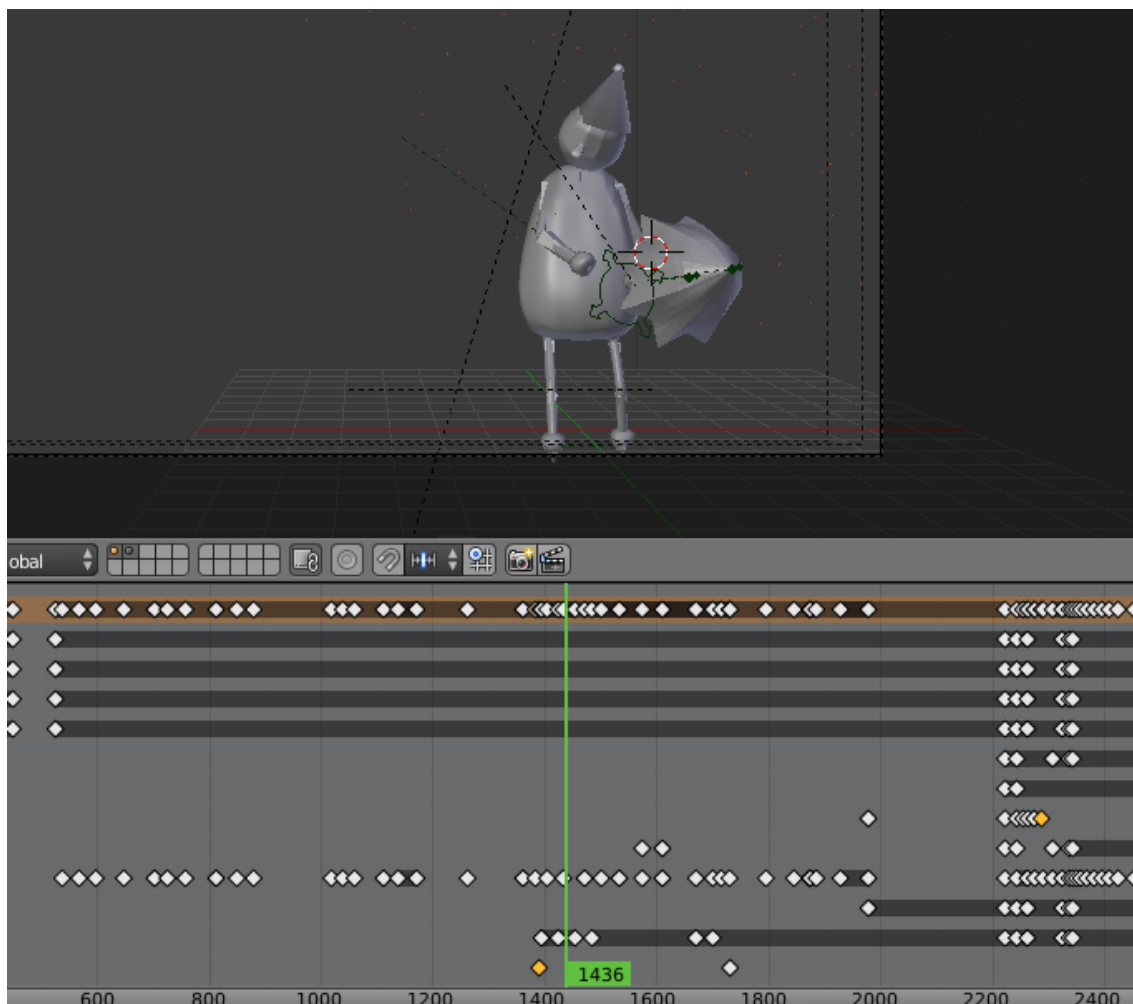
IK-kontrollia varten lisäsin kaksi ylimääräistä luuta, yhden jalkaterään ja yhden polven eteen. Jalkaterän luuta eli IK:n kontrolliluu liikuttamalla on mahdollista liikuttaa koko jalkaa liikuttamalla vain yhtä luuta. Polven edessä ilmassa leijuva luu on IK:n pole-kulmaa kontrolloiva luu. Tätä luuta liikuttamalla voidaan vaikuttaa jalan suuntaan. (Kuva 35)



Kuva 35. Kuvassa punaisella ympyröity pole target-luu joka kontrolloi jalan suuntaa. Pole target-luuta liikuttamalla liikkuu jalan polvi haluttuun suuntaan. Kuvan tapauksessa katsojan näkökulmasta oikealle.

3.3 Animointi

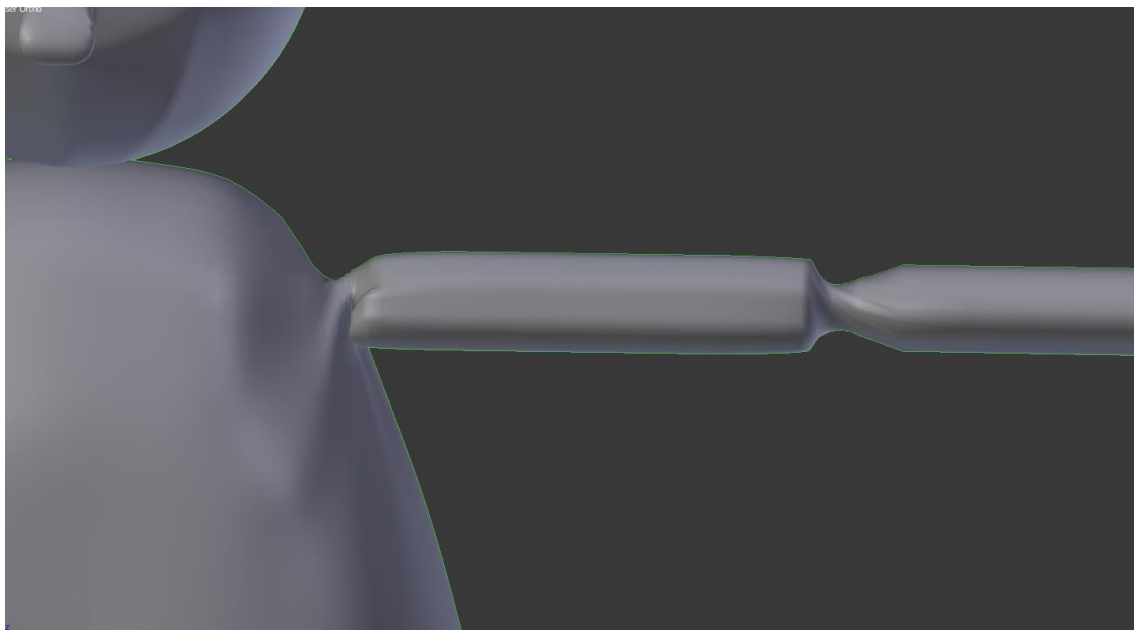
Aloitin animaation tekemisen luonnostelemalla animaation niin, että ajoitin ja animoin ensin kaikki isot liikkeet käsikirjoituksen mukaan. Kun animaatio oli kokonaan karkeasti animoitu aloin lisäämään jo luomieni key framejen väliin uusia key frameja ja säätämään hahmon liikkeitä ja liikeratoja luonnollisemmiksi sekä sulavammiksi. (Kuva 36)



Kuva 36. 3D-näkymä animaatiosta ja animaation key frameja.

3.3.1 Ongelmat käsien kanssa

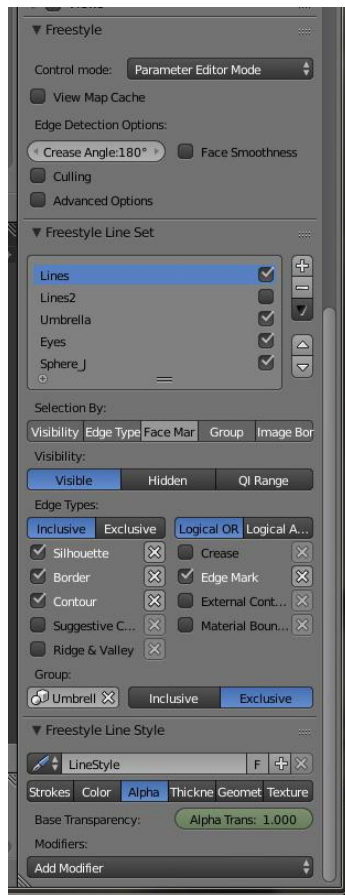
Kuten alussa mainitsin, en tehnyt käsiin IK-kontrollereita ja kaduin sitä lähes koko animointiprosessin ajan. Ensinnäkin käsien animoimiseen meni noin kaksinkertainen aika verrattuna jalkojen animoimiseen, koska jouduin liikuttamaan kädet aina luv kerrallaan haluamaani asentoon. Lisäksi FK-menetelmässä ei ole pole target-luita, mitkä estäisi käden kiertymisen. Tästä syystä huomasin tämän tästä, että liikutellessani hahmon kättä olen vahingossa onnistunut saamaan käden ns. "kierrokselle." (Kuva 37)



Kuva 37. Huolimattoman animoimisen johdosta kiertynyt käsi.

3.4 Renderöinti

Jotta sain animaation ulkoasun halutuksi, minun piti ottaa melkein kaikki freestylen ominaisuudet käyttöön. Tein useamman line setin eri asetuksilla, nämä line setit päällekkäin renderöimällä sain kaikki haluamani viivat näkyviin. Kuten kuvasta 38 voi nähdä, tein viisi line settiä joista neljä oli käytössä. Line seteistä kolme oli hahmon eri yksityiskohtia varten ja yksi sateenvarjolle. Line style – asetuksista säädin pelkästään viivojen väriä ja hieman läpinäkyvyyttä.



Kuva 38. Freestyle asetukset renderöinnille.

Kuvasta 39 voi nähdä, miten viivan tunnistus ei toimi hahmon jokaisessa kohdassa aivan moitteettomasti. Viiva katkeaa tai puuttuu joistakin kohdista ja näyttää hieman epäselvältä. Onneksi pienet virheet ei erotu niin selkeästi liikkuvasta kuvasta. Lisäksi animaatio toisettiin matala resoluutioisella led-näytöllä, minkä takia virheet olivat vielä heikommin havaittavissa.



Kuva 39. Yksi animaation ruuduista renderöityinä. Pysäytetystä kuvasta näkee hyvin, miltä osin freestyle ei toiminut täysin moitteettomasti.

LÄHTEET

- Blender homepage. n.d.a. About
<https://www.blender.org/about/> viitattu 26.1.2016
- Blender homepage. n.d.b. License. Viitattu 26.1.2016
<https://www.blender.org/about/license/>
- Sourceforge freestyle. 2005. Freestyle homepage. Viitattu 23.11.2015
<http://freestyle.sourceforge.net/>
- Blender Manual. n.d.a. Line set. Viitattu 28.1.2016
https://www.blender.org/manual/render/freestyle/parameter_editor/line_set.html
- Blender Manual. n.d.b. Line style. Viitattu 29.3.2016
https://www.blender.org/manual/render/freestyle/parameter_editor/line_style.html
- Blender Manual. n.d.c. Inverse kinematics. Viitattu 31.3.2016
https://www.blender.org/manual/rigging/posing/inverse_kinematics.html
- Blender Manual. n.d.d. Keyframes. Viitattu 31.3.2016
<https://www.blender.org/manual/fr/animation/keyframes/introduction.html>
- Blender Manual. n.d.e. Compositing. Viitattu 31.3.2016
<https://www.blender.org/manual/fr/compositing/introduction.html>
- Blender Wiki. 2013a. Freestyle Line Rendering Engine. Viitattu 23.11.2015
http://wiki.blender.org/index.php/Dev:Ref/Release_Notes/2.67/Freestyle
- Blender Wiki. 2016b. Box Modeling. Viitattu 29.3.2016
https://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Box_Modeling
- Blender Wiki. n.d.c. Rigging. Viitattu 29.3.2016
https://wiki.blender.org/index.php/Doc:2.4/Tutorials/Animation/BSoD/Character_Animation/Rigging
- Blender Wiki n.d.d. Inverse kinematics. Viitattu 29.3.2016
https://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Advanced_Tutorials/Advanced_Animation/Guided_tour/Const/ik
- Blender Wiki. n.d.e. Compositing. Viitattu 31.3.2016
https://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Compositing

Blender Wiki n.d.f. Nodes Viitattu 1.4.2016
https://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Nodes

About Tech. n.d.a. Box Modeling. Viitattu 29.3.2016
<http://3d.about.com/od/Glossary-B/g/Box-Modeling.html>

About Tech. n.d.b. Topology in 3D animation. Viitattu 29.3.2016
<http://3d.about.com/od/Glossary-T/g/Topology.html>