

Henri Leinonen

# 3D-hahmon mallinnus, teksturointi ja animointi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinööriytyö

4.1.2015

Tekijä(t) Otsikko	Henri Leinonen 3D-hahmon mallinnus, teksturointi ja animointi
Sivumäärä Aika	31 sivua + 2 liitettä 4.1.2015
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Tietoverkot
Ohjaaja(t)	Yliopettaja Janne Salonen
<p>Insinööriyön tarkoituksena oli perehtyä 3D-mallinnukseen tekemällä hahmo ja luoda selkeä kuvaus hahmon teon prosesseista ja vaiheista. Työssä esitellään hahmon luonnin vaiheita sekä tekniikoita ja pyritään selventämään, mitä tulee ottaa huomioon tietyissä vaiheissa ja minkä takia. Työssä pyritään myös kuvaamaan teksturoinnin vaikutusta sekä ottaa huomioon sen vaikutukset lopputulokseen. Opinnäytetyö käsittelee hieman myös teoriaa 3D-mallinnuksesta, animoinnista, sekä mitä ohjelmia käytettiin työssä.</p> <p>Vaiheiden etenemistä pyrittiin esittämään kuvien ja tekstin keinoin, jotta koko prosessin kokonaiskuva säilyisi ja hahmottuisi työn edetessä. Ensimmäisenä käytiin läpi hahmon suunnittelua, jonka jälkeen hahmo mallinnettiin, teksturoitiin, skinnattiin, rigattiin sekä animoitiin.</p> <p>Opinnäytetyön tuloksena oli 3D-mallinnettu ja animoitu hahmo, jonka pystyy viedä pelejä varten ohjelmoitavaksi. Työssä opittuja taitojen varassa voin jatkossa malintaa monimutkaisempia kokonaisuuksia ja kehittää työskentelytapoja.</p>	
Avainsanat	3D, mallinnus, vertex, polygon, animointi

Author(s) Title	Henri Leinonen Modelling, texturing and animation of a 3D-character
Number of Pages Date	31 pages + 2 appendices 4 January 2015
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Specialisation option	Data networks
Instructor(s)	Janne Salonen, Principal Lecturer
<p>The thesis showcases the steps and techniques of creating a 3D character and it aims to clarify what needs to be taken into consideration in each step of the way. Also the thesis tries to describe the effects of texturing and its impacts on the final result. The theoretical part describes 3D modeling overall, animation and what programs were used for this project.</p> <p>To understand the whole process and to be able to follow the different steps through the thesis, each step was presented with pictures and text. The thesis covers the basics of planning, designing, modeling, texturing, skinning, rigging and animating a character.</p> <p>The result of this thesis was fully 3D modeled and animated character that could be implemented to games. The thesis is useful for anyone interested in getting involved in games and creating 3Dcharacters.</p>	
Keywords	3D, modeling, vertex, polygon, animation

## Sisällys

Sanasto	5
1 Johdanto	1
2 3D	2
2.1 3D-mallinnus	2
2.2 Animointi	3
2.3 3D-mallinnustavat ja -tekniikat	3
2.4 Ohjelmistot	7
3 Hahmon toteutus	8
3.1 Suunnittelu ja luonnostelu	8
3.2 Hahmon mallinnus	10
3.3 Hahmon teksturointi	16
3.3.1 Bitmap	17
3.3.2 UV mapping ja unwrap	18
3.3.3 Bump map	21
3.4 Hahmon riggaus	22
3.5 Hahmon skinnaus	23
3.6 Hahmon animointi	25
4 Loppuesittely ja mietteet	29
4.1 Testaus	29
4.2 Loppumietteet	30
Lähteet	32
Liitteet	
Liite 1. Texture map	
Liite 2. Bump map	

## Sanasto

Skinnaus	Luiden vaikutusalueiden muokkaus.
Vertex	Piste, jolla on koordinaatit kolmiulotteisessa avaruudessa.
Edge	Reuna, joka erottaa facet ja yhdistää vertexit.
Face	Pinta, joista kolmiulotteinen malli koostuu.
Riggaus	Luiden luonti hahmolle.
Keyframe	Animointiin liittyvä hetki aikajanalla, jossa on tietyt asetukset hahmon eri osille.
Polygoni	Monikulmio, koostuu pisteistä, reunoista ja pinnasta.
UVmap	Prosessi, jossa luodaan 2D kuva kolmiulotteisen kuvan pinnasta.

## 1 Johdanto

Opinnäytetyössä käydään läpi 3D-mallinnukseen liittyviä asioita, termejä ja toteutustapoja sekä luodaan hahmo, jota voidaan myöhemmin käyttää peleissä tai muissa tarkoituksissa. Hahmon luonnissa keskitytään sen toteutuksen tekniikkaan ja sen luonnin vaiheisiin.

Ensimmäisenä käydään läpi yleisesti 3D-mallinnusta sekä animointia pintapuoleisesti sekä esitellään erilaisia tekniikoita, joita käytetään myöhemmässä vaiheessa itse hahmon tekoon. Näiden lisäksi käydään läpi ohjelmistoja, joilla vastaavia töitä voi toteuttaa. Tämän jälkeen käydään läpi hahmon luontiprosessi aloittaen sen suunnittelusta, jonka jälkeen hahmo mallinnetaan, skinnataan, rigataan sekä animoidaan. Tämän jälkeen viedään hahmo Unity-ohjelmaan lopputestausta varten. Opinnäytetyön tarkoituksena on siis luoda kattava katsaus pelihahmon toteutukseen ja siihen, mitä eri työvaiheissa tulisi ottaa huomioon.

## 2 3D

### 2.1 3D-mallinnus

Aiemmin eniten 3D-mallinnusta on käytetty arkkitehtien ja suunnittelijoiden piirissä, mutta nykyään tietotekniikassa, elokuvissa, peleissä ja teollisuudessa tulee joka päivä vastaan jotain, joka on suunniteltu ja toteutettu 3D-mallinnuksen prosesseja käyttäen. Näiden lisäksi 3D-tulostus on kasvava teollisuuden ala, jossa tulostin siis tulostaa ja kaivertaa tietokoneella mallinnetun välineen, työkalun tai hahmon oikeaksi malliksi. Näitä palveluita on ollut jo 1980-luvulta lähtien, mutta ala on saanut vasta nyt suurta huomiota, sillä sitä on nyt suunnattu myös kuluttajien käyttöön.

3D-mallinnuksella tarkoitetaan siis prosessia, jota käytetään kolmiulotteisen mallin luontiin erityistä mallinnusohjelmaa käyttäen. Tässä prosessissa luodaan pisteitä kolmiulotteisessa koordinaatistossa, joiden sijainti määräytyy sen x-, y- ja z-koordinaattien mukaisesti. Tämä kolmiulotteisessa avaruudessa sijaitseva piste esitetään matemaattisesti, mallinnusohjelma laskee ja esittää sen grafiikkana ohjelman käyttäjälle [1]. Tätä laskentaa ja toimintaa kutsutaan ns. real-time render engineksi, eli reaaliajassa päivittyväksi mallinnusmoottoriksi, joka esittää määrättyjen pisteiden ja niistä luotujen mallien muotoa. Tällainen mallinnusmoottori on käytössä suurimmassa osassa nykypäiväisiä 3D-ohjelmia.

3D-mallin luontiprosessissa aloitetaan yksinkertaisesta esineestä (primitive), jota kasvatetaan ja muokataan monimutkaisemmaksi ja yksityiskohtaisemmaksi. Alkeelliset kohteet voivat olla yksittäisiä pisteitä (vertex), kaksiulotteisia viivoja (edge) tai kolmiulotteisia kappaleita (face). [2.] Vertex on siis piste kolmiulotteisessa avaruudessa, edge on kahden vertexin yhdistämä viiva tai reuna. Näiden yhdistämää aluetta kutsutaan polygoniksi. Käyttäen erilaisia ohjelmiston tarjoamia tapoja voidaan alkeellisista esineistä muodostaa erilaisia malleja. Mallinnustapoja ja -tekniikoita on useita ja mallin luojan tulisi olla tietoinen eri tavoista, sillä jokaisella tavalla ja tekniikalla on hyvät ja huonot puolensa.

## 2.2 Animointi

Animoinnilla tarkoitetaan elokuvan toteutusta kuva kovalta. Animaatiot toteutetaan piirtämällä jokainen kuva erikseen, tai muokkaamalla edellistä, ja tämän jälkeen kuvat esitetään peräkkäin, joka luo illuusion liikkeestä [3]. Animaatioiden luonti on hyvin pitkä prosessi, sillä yhden sekunnin aikana nykyään elokuvissa on 24-30 kuvaa. Vaikka piirretyt animaatiot näihin kuvamääriin sekunnissa eivät yllä, voi jokainen kuvitella työn määrän edes minuutin pituisen animaation tekoon. Nykyään animaatiot tehdään suureksi osaksi tietokoneen avulla, joka tietenkin lyhentää ja vähentää työn aikaa ja määrää.

Vaikka ala on jo suhteellisen pitkään ollut olemassa, harmiksemme animoinnin ja 3D-mallinnuksen alalla ei kaikki vielä ole täysin standardoitu, ja eri valmistajilla ja tekijöillä voi olla useita merkityksiä ja termejä samoille asioille. Tämä tietenkin hankaloittaa asioiden tutkimista ja saattaa myös harhauttaa näistä tietoa etsivää. Suomennoksia läheskään kaikkiin termistöihin ei ole valmiina, joten insinööriö sisältää paljon englanninkielistä termistöä.

Insinööriö loppupuolella käydään läpi animointia ja mitä siinä tulisi ottaa huomioon, kun tekee vastaavanlaista hahmoa mm. pelikäyttöön. Työn lopussa mallinnettu hahmo animoidaan ja sille luodaan liikeradat käyttäen yksinkertaisia luita sekä niistä koottua luurankoa.

## 2.3 3D-mallinnustavat ja -tekniikat

On olemassa automaattisia ohjelmistoja ja välineitä, joilla voidaan 3D-esineitä hahmottaa. Näiden avulla voidaan luoda kolmiulotteisia esineitä, työkaluja tai muita malleja todella nopeasti ja vaivattomasti. Yhtenä esimerkkinä voidaan esitellä 3D-skanneri, joka kerää tietoa esineen muodosta ja väristä. Tätä tietoa voidaan käyttää hyväksi ja luoda sen avulla automaattisesti ohjelmiston avulla kolmiulotteinen malli, jota voidaan käyttää eri tarkoituksiin.

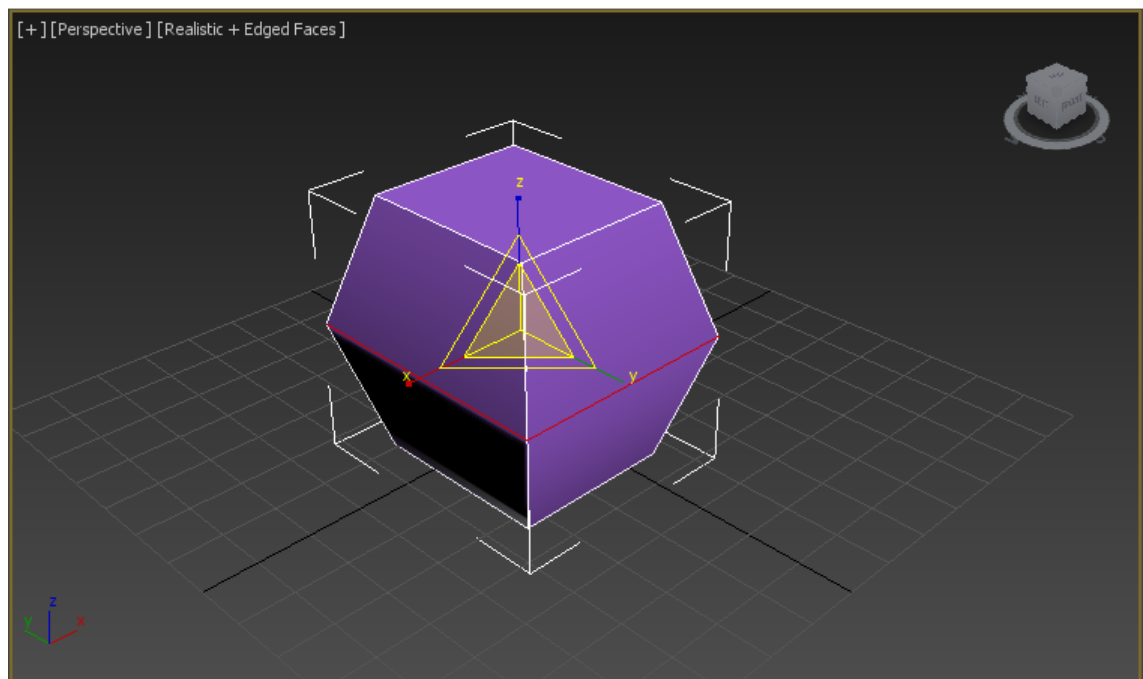
Toisaalta on laitteita, kuten XBOX-kinect, joka on suunniteltu pelejä varten. Laite sisältää kaksi kameraa, jotka seuraavat käyttäjän liikkeitä reaaliajassa. Ohjelmiston ja kameroiden yhteistyön tuloksena se luo 3D-hahmon, jonka liikkeitä voidaan käyttää mm. peleissä ja liikkeentunnistuksessa. Automaattisissa mallinnustavoissa hyvänä



puolena on se, että se on paljon nopeampaa kuin käsin tehtynä. Huonona puolena voidaan pitää sitä, että vaikka skannaus ja muut tavat tulevat tarkemmiksi koko ajan, joudutaan usein luotuun hahmoon tehdä korjauksia käsin.

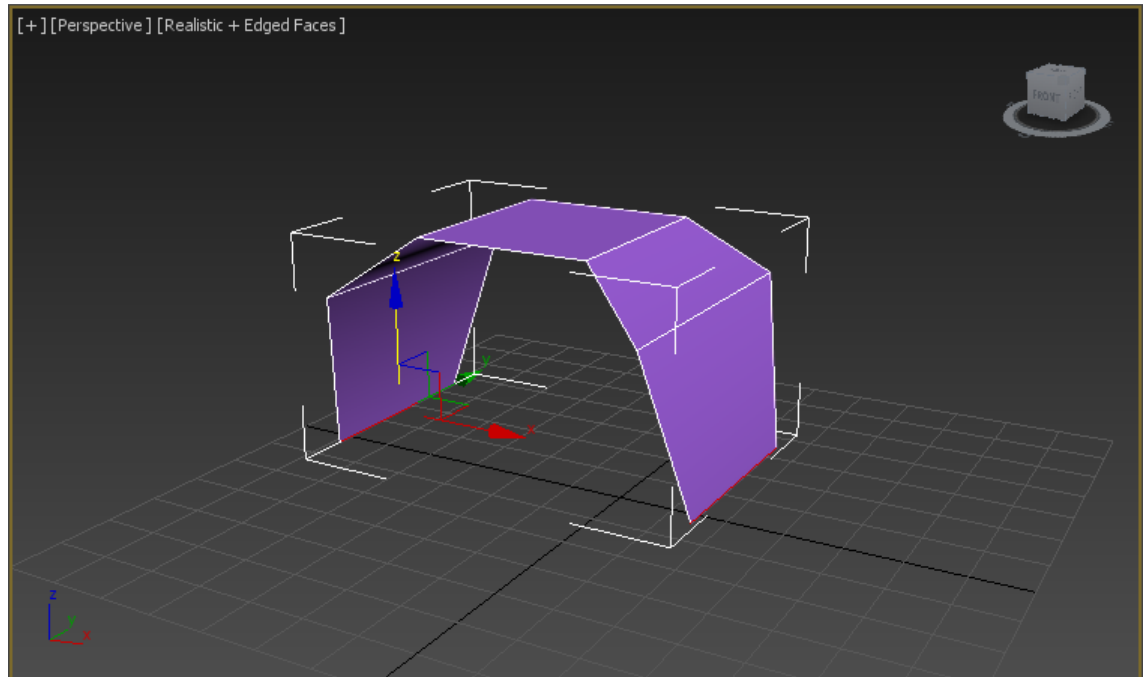
Automaattisen mallinnuksen lisäksi on käsin luotujen mallien teko, johon insinööriyössä siis perehdymme. Yleisesti mallinnukseen käytetään kolmea eri tapaa, joita seuraavaksi käsittelemme. Hyviä suomenkielisiä käännöksiä tekniikoille ei löytynyt, joten näissä on pysytty englanninkielisissä termeissä.

Käytetyin mallinnustekniikka on ns. Box Modeling. Tässä tekniikassa lähdetään liikkeelle alkukantaisesta kappaleesta, kuten laatikosta, johon lisätään yksityiskohtia jakamalla sen alueita pienempiin osiin ja venyttämällä eri alueita. Tämä tapa on ehkä nopein ja yksi tärkeimmistä tavoista osata mallintaa. Huonona puolena tässä kuitenkin on se, että mallia pitää muokata paljon, ja se on vaikea luoda animointia varten. Tätä tekniikkaa kuitenkin käytämme hahmon luonnissa myös insinööriyön käytännönvaiheissa.



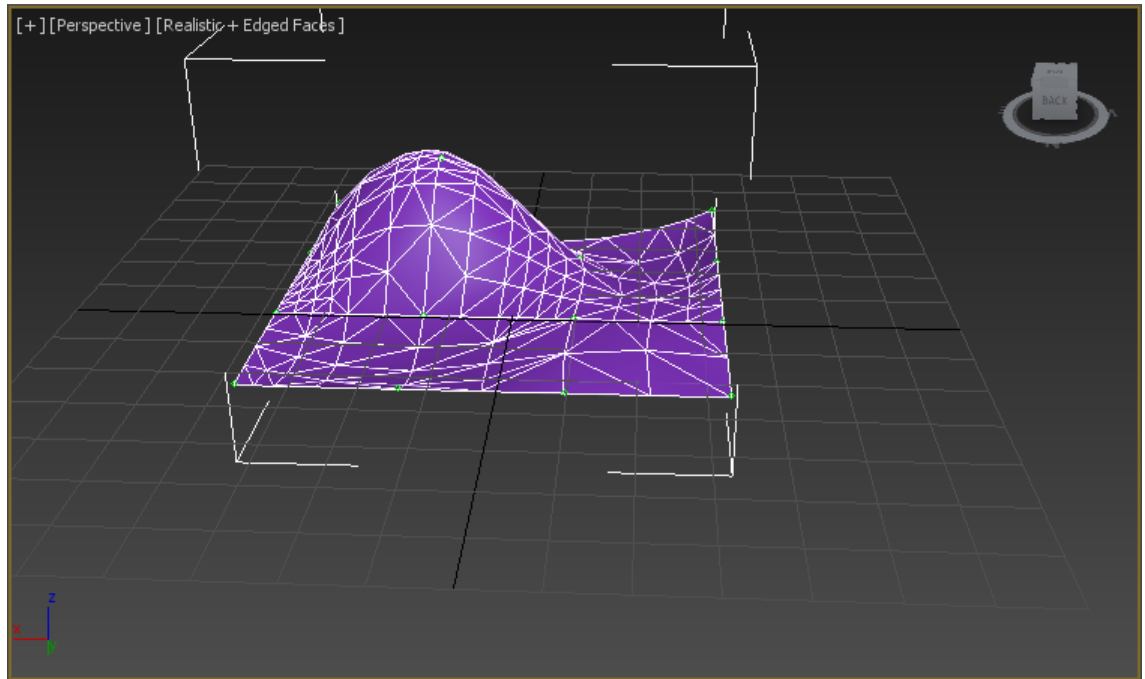
**Kuva 1. Box Modeling -tekniikka**

Toinen käytetty tekniikka on polygon modeling/edge extrusion -tekniikka. Tekniikassa yleisesti aloitetaan yhdestä nelikulmaisesta palasta, jonka jälkeen yhdestä sivusta venytetään seuraava alue. Kun tätä samaa toimenpidettä toistetaan tarpeeksi monta kertaa, saadaan luotua haluttu malli hyvin tarkasti. Vaikka tämä ei olekaan nopein tapa luoda hahmoa, on se yksi tarkimmista tavoista ja sen takia yksi käytetyimmistä. Tällä tavalla voi myös saada parempia tuloksia aikaan animoinnin suhteen, koska mallia voi muokata ja suunnitella alusta alkaen tarpeittensa mukaisesti.



**Kuva 2. Edge extrusion -tekniikka**

Viimeinen esitelty tapa on Non-uniform rational B-spline -tekniikka, eli yleisesti NURBS. Tämä tekniikka on yksi parhaista luomaan sileitä kaarevia pintoja. Toisin kuin ylläesitellyt tavat, NURBS-tekniikassa, luodaan yksittäisiä pisteitä, jotka ohjaavat sen ympäristössä olevan pinnan liikkeitä suhteessa muihin pisteisiin. Tämä tekniikka ei sovellu kaikkiin käyttötarkoituksiin, mutta koska se on hyvin kevyt käyttää, se sisältää sen muotoihin nähden suhteellisen vähän dataa ja sen avulla pystyy luomaan sileitä pintoja, on sen käyttö hyvin yleistä varsinkin pelien käytössä.



**Kuva 3. NURBS-tekniikka**

Tekniikan valitsemiseen vaikuttaa hahmotettava malli, sen käyttö sekä mallintajan omat kokemukset ja tottumukset eri tekniikoiden välillä. Tekniikoita myös saatetaan käyttää yhdessä ja toistensa päällä parhaan lopputuloksen saavuttamiseksi. Esimerkiksi Box modeling -tekniikkaa voidaan käyttää hahmon suurimpien pintojen, kuten kehon ja raajojen luontiin, ja tämän jälkeen käyttää Edge extrusion -tekniikkaa sormien tai varpaiden luontiin.

## 2.4 Ohjelmistot

3D-mallinnukseen tarkoitettuja ohjelmistoja on nykyään hyvin paljon. Ohjelmistot vaihtelevat ilmaisista ohjelmistoista kalliisiin ammattiohjelmistoihin, mutta viime vuosina näiden ero on kuroutunut yhteen todella nopeasti, ilmaiset ohjelmat ovat nousseet monien suosioon jopa maksullisten ohjelmistojen yli. Kuluttajille suunnatut ilmaiset ohjelmistot ovatkin ottaneet toisen suunnan kehityksessä ja pyrkivät saamaan tulosta mm. lisäpalveluiden ja valmiiksi mallinnettujen hahmojen/objektien luonnilla. Tästä syystä myös kalliimpien ohjelmistojen tuottajat ovat joutuneet jakamaan ilmaisversioita rajoitetuin toiminnoin tietysti.

3D-ohjelmista yksi suosituimmista on tällä hetkellä Blender, joka on ainoa ilmainen ammatilaiskäytössä oleva mallinnus ja animaatio-ohjelma. Myös Blender on tullut vuosien mittaan kuluttajaläheisemmäksi ja sitä on muokattu käyttäjäystävällisemmäksi. Koska Blender on ilmainen, sitä itsekkin käyttäneenä, voin todeta, että ilmaiseksi ohjelmistoksi se on loistava. Vaikka Blender muuttuu paljon, jolloin erilaiset kurssit tai videot ohjelmasta vanhentuvat turhan nopeasti, on se hyvin kattava ohjelmisto. Tämän lisäksi siihen on tarjolla paljon maksuttomia lisäosia ja materiaalia, jota voi hyödyntää.

Blenderin toisena ääripäänä voidaan pitää 3ds Maxia, joka on yksi käytetyimmistä kaupallisen puolen ohjelmistoista. Ohjelmisto maksaa itsessään jo paljon, jopa 180 \$ kuukaudessa, mutta sillä on suuri käyttäjäryhmä ja ongelman ilmetessä rahoille saa vastinetta. Se on hyvin laaja mallinnusohjelma ja on erityisen käytetty videopelien keskuudessa. 3ds Max on myös nykyisin helppokäyttöinen ja helposti lähestyttävä, jos rahasta ei ole pulaa. Tämä on ehdottomasti yksi parhaista 3D-mallinnusohjelmista.

Kolmen käytetyimmän 3D-ohjelmiston joukkoon voidaan myös laskea Autodeskin Maya ja sen kevyempi versio Maya LT, joka on suunnattu enemmän pelien kehityksen puoleen [4]. Vaikka käyttöliittymä ei ole aivan 3ds Maxin tasoa, sen toiminnot ja yksinkertaisuus ovat valttia. Kaikki tarvittava pelinkehitykseen löytyy ja helppo hahmojen ohjelmistosta vienti tekee siitä loistavan ja helpon työkalun. Myös hinta ei ole 3ds Maxin tasoa, vaan noin 30 dollarin kuukausiluokkaa. Maya ja Blender on erityisesti Indie-pelien tekijöiden suosiossa tällä hetkellä.

Vaikka ohjelmistojen välillä on eroavaisuuksia, niin kaikissa näistä kolmesta ohjelmistossa on kaikki, mitä insinööriyössä tehtävän hahmon luontiin tarvitaan. Ainoastaan oma mieltymys, ja se, haluaako siitä maksaa, vaikuttaa valintaan.

Insinööriyössä 3D-mallinnusta varten tarvitsin vain kahta ohjelmistoa. Itse mallinnukseen käytin 3ds Maxia mallinnukseen ja Adobe Photoshopia hahmon teksturointiin. Tämän jälkeen käytin Unity-pelimootoria animaatioiden testaukseen.

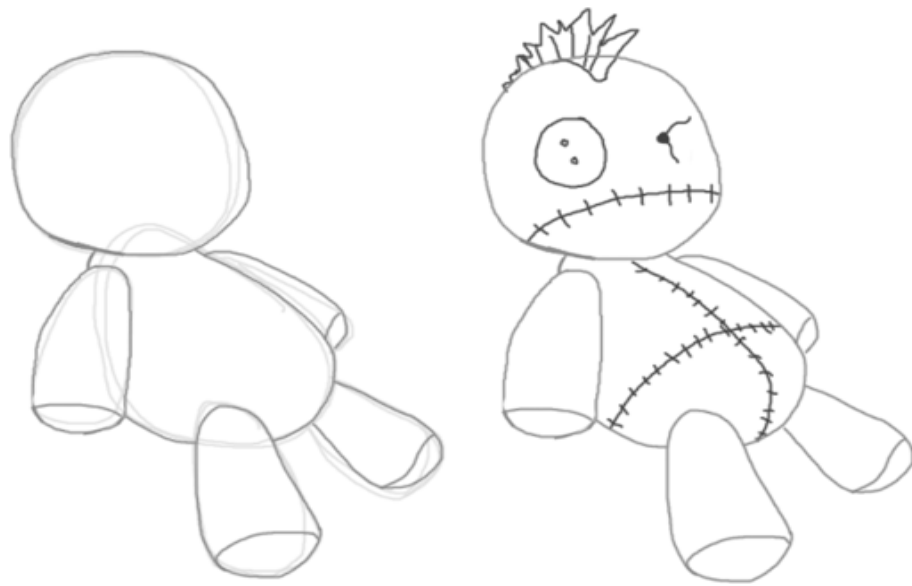
### **3 Hahmon toteutus**

Hahmon toteutukseen kuuluu paljon eri osa-alueita, niiden ymmärtäminen on tärkeä osa prosessia luoda kokonainen hahmo. Koska insinööriyön hahmo ei ole osa mitään suurempaa kokonaisuutta, kuten peliä tai elokuvaa, sille ei tarvitse keksiä tähän käyttötarkoitukseen taustoja tai ominaisia tapoja tehdä asioita. Yleisesti saman hahmon toteutuksessa on mukana useita henkilöitä, jotka vastaavat kukin oman työvaiheen toteutuksesta. Hahmon toteutuksessa lähdetään liikkeelle suunnittelusta.

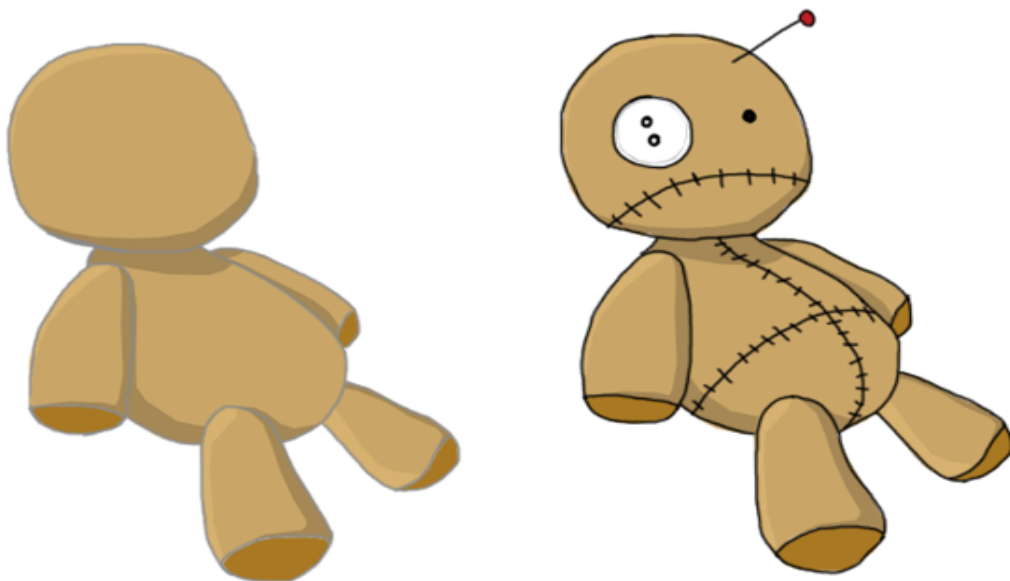
#### **3.1 Suunnittelu ja luonnostelu**

Suunnitteluun kuuluu siis kaikki hahmoon liittyvä, joka on mahdollista ennalta toteuttaa, kuten esimerkiksi käsikirjoitus ja hahmon ominaispiirteiden kartoitus. Toisaalta kuten kaikessa sanotaan, hyvin suunniteltu on jo puoliksi tehty. Vaikka tässä on jotain totuutta, tämä ei suoraan päde 3D-hahmon mallintamiseen, vaan suuri osa aikaa kuluu hahmon toteutukseen.

Hahmon piirteiden ja tapojen kartoituksen jälkeen varsinainen työ aloitetaan hahmon luonnostelusta, jossa hahmosta olevat ideat siirretään artistille, joka luo piirroksia tai konseptitaidetta hahmosta ja sen ympäristöstä. Ilman taustatietoja hahmosta artistin on hyvin vaikea alkaa luonnostelemaan, mitä ja minkälaista kuvaa halutaan hahmosta esittää katsojalle ja käyttäjille. Hahmoksi päätettiin toteuttaa voodoo-nukke, koska uskon sen olevan tarpeeksi monimutkainen tähän työhön ja vastaavanlaista hahmoa en vielä ole itse luonut. Ensimmäiseksi luonnosteltiin kuvia mahdollisesta ulkonäöstä Adoben Photoshopilla.



**Kuva 4. Hahmon luonnostelua 1**



**Kuva 5. Hahmon luonnostelua 2**

Ilman kaksiulotteisia kuvia on hyvin hankalaa alkaa hahmottelemaan hahmoa kolmiulotteisessa ympäristössä. Kuvat siis auttavat jo tällaisina mallintajan toimia, sillä on jotain johon hahmoa verrataan. Yleisesti hahmoista tehdään ns. sivu- ja etukuvat, joiden perusteella voidaan 3D-hahmoa muokata ja toteuttaa helposti edge extrusion -tekniikan avulla. Tällöin voidaan siis tehdä hahmon ulkomuotoja seuraamalla näitä

kuvia. Toisaalta kuvien tulisi olla tarkkoja ja vastata toisiaan, jos niitä haluaisi täysin seurata mallintamisen vaiheessa. Uskon kuitenkin näiden kuvien riittävän tätä toteutusta varten hyvin. Eri vaiheiden määrittäminen ja selkeä rajaaminen on myös tärkeää, jotta projektit etenevät oikeaan tahtiin ja jokainen tietää, mitä osaa on toteuttamassa ja millä aikataululla. Selkeät vaiheet myös siirtyvät 3D-hahmon toteutukseen, jota käydään siis seuraavaksi läpi.

### 3.2 Hahmon mallinnus

Ennen mallinnuksen aloitusta on hyvä olla tietoinen muutamasta asiasta. Ensimmäinen tarvittava tieto, tuleeko tämä hahmo pelikäyttöön, on hyvin olennainen. Tällä haetaan takaa siis sitä, että pyritään tekemään mahdollisimman tarkka hahmo mahdollisimman vähällä määrällä polygoneja, koska mitä enemmän polygoneja on, sitä raskaampi hahmo on pyörittää peleissä. Tämä valinta käytännössä perustuu siihen, että peleissä on käytössä kolmen vertexin tekniikka yhtä facea kohden, jolloin vertexien ja polygonien määrä jää lopputuloksessa pienemmäksi ja hahmo on "kevyempi". Jos tätä hahmoa ei tule pelikäyttöön, voidaan hyvin käyttää neljän vertexin tekniikkaa, jolloin saavutetaan luonnollisemman näköisiä tuloksia.

Toinen seikka, joka on huomioitu jo mahdollisessa referenssikuvassa, on asento, johon hahmo mallinnetaan. T-asennoksi tai referenssi-asennoksi kutsuttu olemus on yleisin asento, johon hahmot mallinnetaan ennen animointia. Tämä asento tarjoaa aloituksen, jossa kaikki raajat ovat suorina ja valmiina helpottamaan animointia. T-asennoksi sitä kutsutaan, koska jalat ovat suorina ja kädet sivusuuntaan avattuina [5]. Asento auttaa myös mallinnuksen jälkeisessä teksturointivaiheessa, jossa hahmolle lisätään tekstuurit.

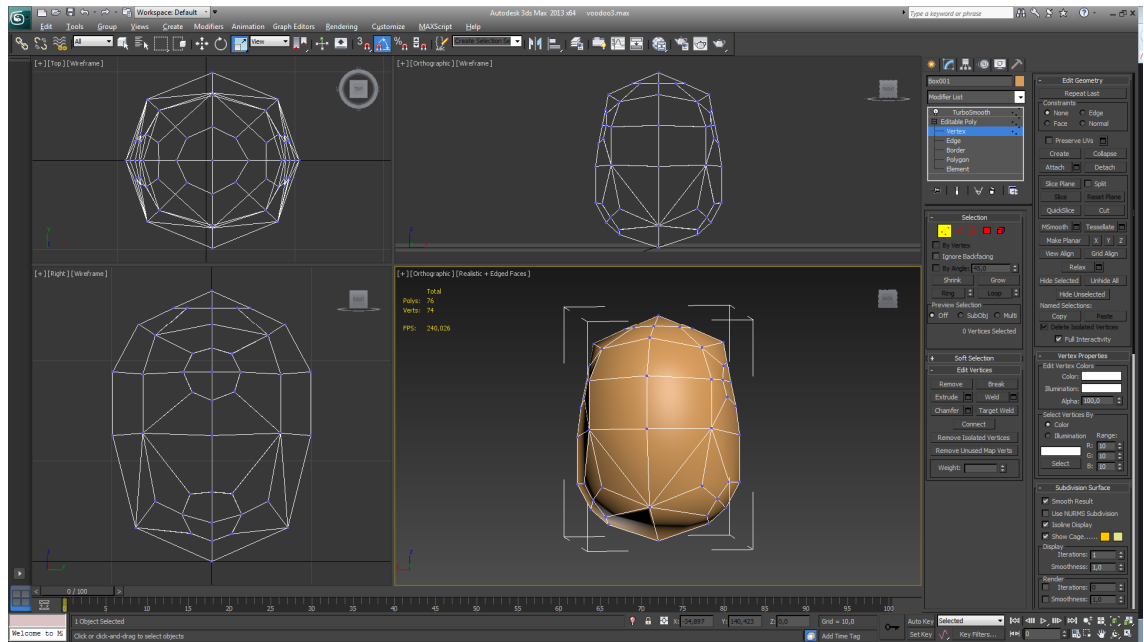
Näiden tietojen jälkeen voidaan valita, millä tekniikalla aloitetaan mallinnus. Itse valitsin box -modeling tekniikan, koska tällä tekniikalla olen myös aiemmin tehnyt hahmoja, ja se on toiminut tähän tarkoitukseen hyvin. Aloituksessa on tärkeää myös huomioida, että hahmo on keskitetty oikein. Asetettuaan hahmon keskelle koordinaatistoa, voidaan käyttää työkalua, joka peilaa toisen puolen suoraan, jolloin ainoastaan tarvitaan hahmotella "puolet" hahmosta. Itse en tätä työssäni käyttänyt, vaikka se olisi nopeuttanut työtä huomattavasti.

Kokemuksen perusteella päätin lähteä liikkeelle vartalon teosta ja keskittyä suurien pintojen luontiin ja jättää yksityiskohdat myöhemmälle ajankohdalle. Box modeling-tekniikassa lähdettiin siis laatikosta liikkeelle, jonka jälkeen laatikko muunnetaan tilaan, jossa sen polygoneja, vertexejä ja edgejä voi luoda ja muokata. Tämä tapahtui objektin valinnan jälkeen oikean hiirennapin painalluksella ja valitsemalla Convert to: - Convert to Editable Poly. Tämän voi myös tehdä Modify-välilehdeltä, josta myös lisäsin Turbo Smooth -määritteen. Turbo Smooth -nimensä mukaisesti pyrkii tasoittamaan luotua massaa lisäämällä siihen reunoja siinä olemassa olevien reunojen väliin. Tällä työkalulla voi nopeasti lisätä reunoja ja tarkentaa kuvaa, mutta on hyvä kuitenkin ottaa huomioon, että polygon- ja vertex-määrät kasvavat nopeasti vastaavanlaisen työkalun avulla. Kun valitsee Editable Poly -määrityksen, Turbo Smooth -vaikutus katoaa. Tämä johtuu siitä, että Editable Poly -määritys on tehty aiemmin. On myös mahdollista käyttää vaihtoehtoa, joka näyttää lopputuloksen koko ajan painamalla Show end result on/off toggle. Itse kuitenkin työskentelen mieluiten ilman Turbo Smoothia, kunnes haluan nähdä lopputuloksen.

Siirtämällä vertexejä, lisäämällä reunoja ja kasvattamalla polygoneja mallin kehon muoto alkaa hahmottumaan. Vartaloa tämän jälkeen tuli muokata siten, että jalat ja kädet on mahdollista luoda helposti ja vaivattomasti. Tämän takia pyrittiin luomaan ulostuloaukot käsille, jaloille ja päälle, jotta työskentely myöhemmissä vaiheissa olisi helpompaa. On myös hyvä huomioida jo tässä vaiheessa, aikooko hahmoa animoida, ja koska työssä tulee olemaan animointia, hahmo on toteutettava siihen toimivalla tavalla. Raajojen lähtöalueiden tulee olla sellaisia, jotka pystyvät venymään siten, että ympärillä olevat vertexit eivät joudu liikkumaan raajan liikkuessa. Tämä toteutetaan yleensä siten, että raajojen ja vartalon väliselle alueelle pyritään luomaan suhteellisen paljon vertexejä, jotka vastaavat liikkeestä animaation aikana. On myös hyvä periaate toteuttaa muu vartalo suoraviivaisesti ja animaatiota huomioon ottaen. Tässä vaiheessa kuitenkin on selvää, että kyseisellä hahmolla ei tule liikettä vartalon alueelle lainkaan, vaan se pysyy paikallaan mahdollisimman paljon.

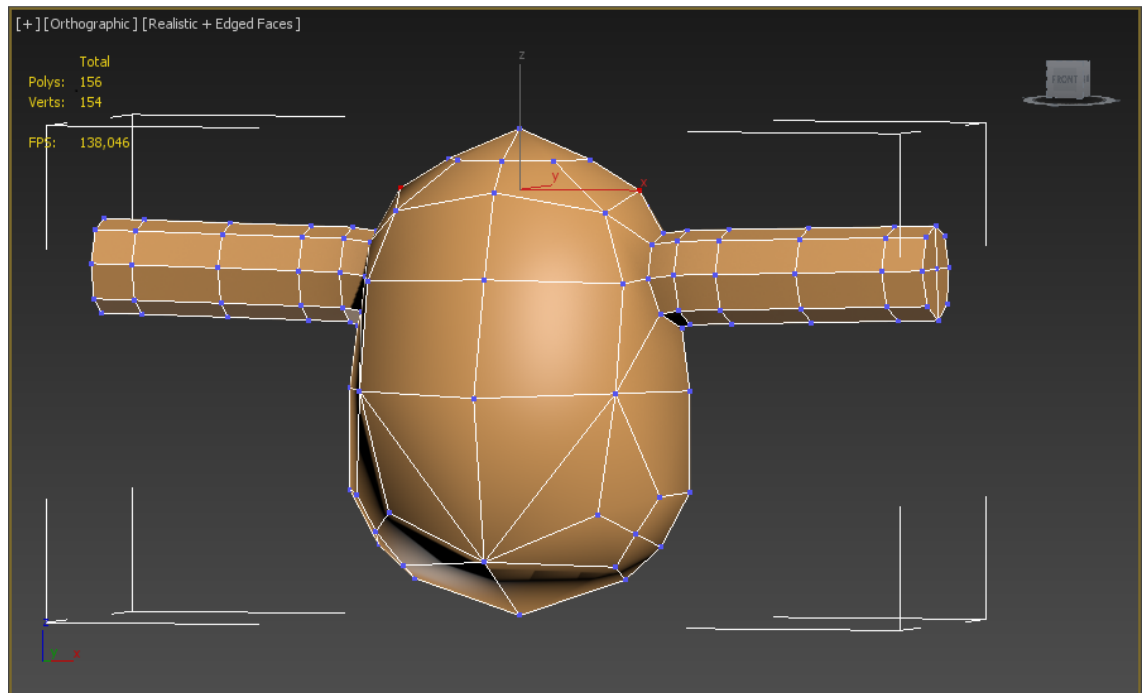
Kuvassaa 6 huomataan, kuinka kaikille raajoille on tehty alue, josta on helppo raajoja alkaa muokkaamaan. Vasemmalla alhaalla olevassa ruudussa tämä näkyy parhaiten, jossa keskimäinen ympyrä kuvaa aluetta, josta kädet tulevat ja alempana oleva on jaloille varattua tilaa.





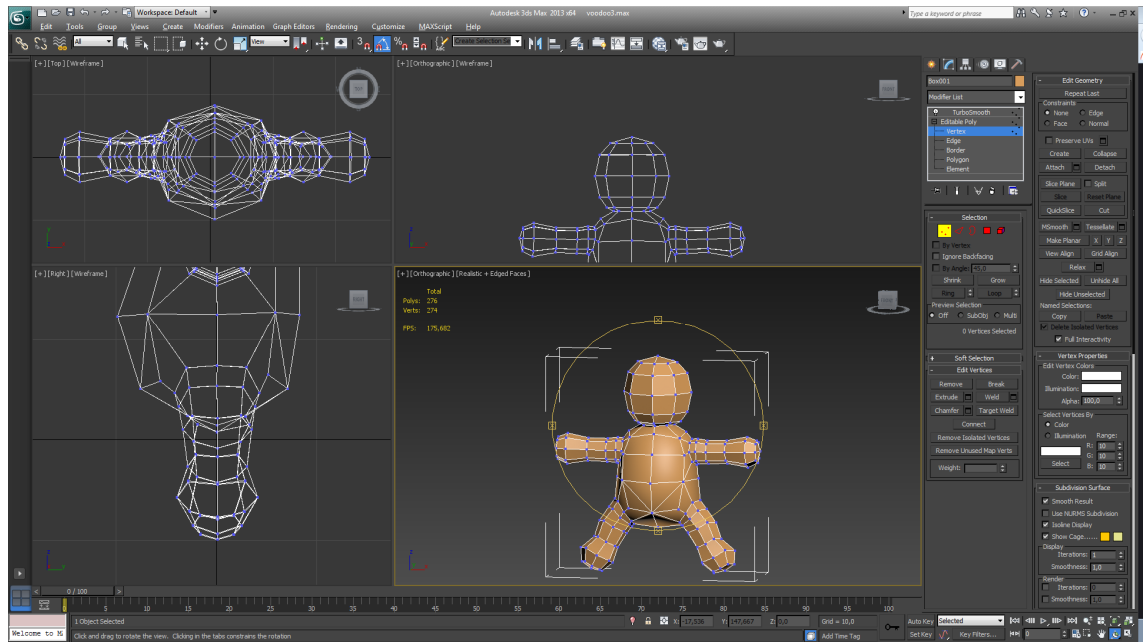
**Kuva 6. Vartalon mallintaminen ja raajojen ulostulojen määrittäminen**

Tämän jälkeen hahmolle aloitettiin tekemään raajoja. Raajojen teko päätettiin käyttämällä extrusion välinettä, jossa valitaan polygon, eli yksi tai useampi alue, jota sen jälkeen venytetään ja luodaan vaihe vaiheelta uusia alueita, vertexejä ja pintoja. Tätä toimenpidettä toistetaan, kunnes haluttu tulos on saavutettu:



**Kuva 7. Käsien luonti käyttämällä extrusion-työkalua**

Tämän jälkeen voidaan sama toimenpide toistaa jaloille ja päälle. Yleisten pintojen tuottamisen jälkeen voidaan siirtyä muokkaamaan niitä ja tehdä niistä luonnollisemman ja alkuperäispiirroksessa olevan mukaisia. Tässä tapauksessa pyrittiin tekemään voodoo-nukke, jonka raajat ovat muuten suhteellisen kapeat ja raajojen päissä on "enemmän täytettä". Tämän pystyi toteuttamaan valitsemalla raajoista reunoja ja valitsemalla Loop-toiminteen, joka valitsee reunan mukaisesti kehän, kunnes palaa reunan toiseen päähän ja muodostaa loopin. Tämä ei kuitenkaan jokaisessa tapauksessa toiminut, vaan joutui käsin valitsemaan reunat yksitellen. Kehän valitsemisen jälkeen voidaan sen kokoa kasvattaa painamalla R, joka on pikanäppäin Scalelle. Kasvattamalla tämän ja muiden ympärillä olevien kehien kokoa voidaan luoda pinnan muotoa hyvin nopeasti. Muita pikanäppäimiä on E, Rotate, jolla kappaletta voidaan pyörittää sekä W, Move, jolla kappaletta voidaan liikuttaa. Pikanäppäimet nopeuttavat työskentelyä huomattavasti ja niiden oppiminen tulee luonnostaan. Extrude-toimenpidettä jatkettiin myös jaloille ja päälle. Pitkän työn jälkeen hahmo alkoi muodostua ja näyttämään luonnoksen mukaiselta.



**Kuva 8. Pään ja jalkojen lisäyksen jälkeinen hahmo**

Hahmon ulkomuoto verrattuna alkuperäiseen luonnokseen oli vielä erilainen, jonka takia keskivartaloa pienennettiin hieman ja päätä suurennettiin. Tällöin pystyttiin todentamaan, että hahmon raajat ovat oikeissa mittasuhteissa toisiinsa nähden ja vastaavat lähes alkuperäistä referenssikuvaa.

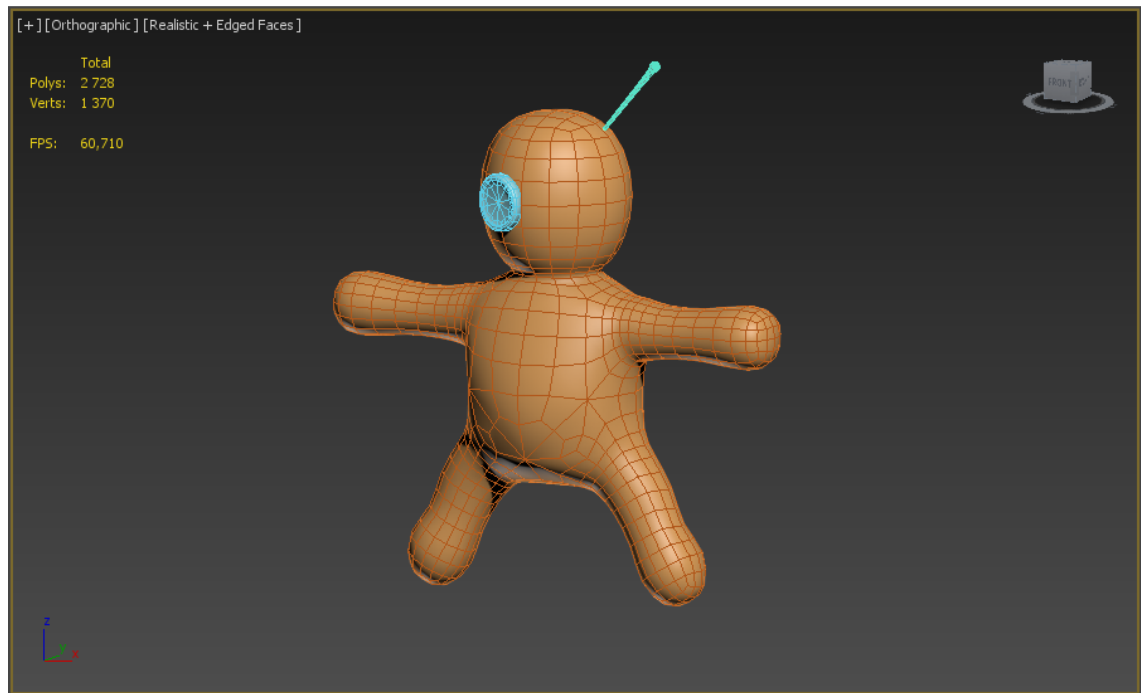
Koska yleiseen olemukseen oltiin tässä vaiheessa tyytyväisiä, pystyttiin yksityiskohtiin alkaa panostaa. Alkuperäisessä piirroksessa nukella oli toisena silmänä nappi ja toisesta silmästä nappi puuttui sekä nukella oli nuppineula päässä. Päätettiin, että nappi sekä nuppineula mallinnettisiin, ja reikä toteutettaisiin tekstuurilla. Haluttiin myös toteuttaa reiät nappiin, sekä lisäämään langanpätkä napin reikien välille tuomaan realistista ilmettä. Napin mallinnus aloitettiin sylinteristä, jonka jälkeen tuli miettiä, miten napissa olevat reiät toteutettaisiin. Reiän avaus aloitettiin siten, että yhdistettiin eri reunoja Connect-työkalulla ja lisäämällä lisää vertexejä näiden välille. Luomalla tarpeeksi vertexejä voitiin reiät avata helposti, mutta yksityiskohtien määrä kuitenkin näkyi huomattavasti polygon- ja vertex-määrissä, joka yleisesti halutaan pitää mahdollisimman alhaisena.

Neulan teko aloitettiin myös sylinteristä, mutta se kasvatettiin pituussuunnassa ja scale-työkalun avulla toiseen päähän luotiin pallo, jotta neula muistuttaisi nuppineulaa erityisesti. Tämän jälkeen molempiin erillisiin objekteihin lisättiin Turbo Smooth

pehmentämään reunoja ja alueita. Tässä mallinnuksen vaiheessa voidaan tarkastella jo, miltä lopputulos tulee näyttämään ja voidaan valita Turbo Smooth näkyviin kaikille objekteille.

Myös tässä vaiheessa oli huomioitava, että neula, nappi sekä napin läpi menevä lanka olivat kaikki eri objekteja, eikä mikään ollut toisiinsa liitettyinä. Tämän sai korjattua tekemällä joko näistä objekteista ryhmiä, valitsemalla ensin halutut objektit ja sen jälkeen Group - Group. Tätä ryhmää pystyi hallitsemaan valitsemalla Group - Open, jolloin objektit olivat jälleen erillään. Group-työkalu on samalla kätevä, mutta kömpelö ja tästä syystä päädyttiin ryhmittämään ainoastaan nappi sekä sen läpi menevä lanka. Näiden tiedettiin olevan toisiinsa nähden aina paikallaan. Tiedettiin myös, että animaatiossa näitä ei tarvitse erikseen animoida, jolloin niihin ei tarvitse erityistä huomiota kiinnittää.

Toinen tapa yhdistää kappaleita on linkata ne toisiinsa. Tämä on hierarkkinen toimenpide, jossa rakennetaan hierarkia ns. child-objektin ja parent-objektin välillä. Tämä tarkoittaa käytännössä sitä, että child-objektina oleva voi liikkua tai skaalautua vaikuttamatta parent-objektiin, mutta jos parent-objektia liikutetaan, myös child-objekti liikkuu samaan tahtiin [6]. Tällä tavalla yhdistin napin nukkeen sekä nuppineulan nukkeen, jolloin nappi ja nuppineula ovat child-objekteja ja niitä voi tarvittaessa liikutella ja skaalata vaikuttamatta itse nukkeen lainkaan. Tämän linkin voi myös purkaa halutessaan, jolloin riippuvuuksia ei enää objektien välillä ole.



**Kuva 9. Mallinnus Turbo Smooth -määrityksen jälkeen**

### 3.3 Hahmon teksturointi

Teksturointi on prosessi, jossa hahmo saa uuden materiaalin iholle eli skinille. Uudelleen teksturointia käytetään ja kierrätetään myös hyvin paljon peleissä, joissa on paljon yksinkertaisempaa tehdä uudelleenvärjäys yhdelle hahmolle kuin tehdä kokonaan uusi hahmo ja uusi "skini". Teksturoinnilla on siis suuri merkitys peleissä, mutta myös elokuvissa, jossa pyritään luomaan mahdollisimman tarkan näköinen ja realistinen ilme. Hyvän ja huonon tekstuurin eron näkee jo pitkälti, ja se on itse mallin kanssa yksi tärkeimpiä vaiheita lopputuloksen kannalta.

Tässä vaiheessa täytyy olla täysin varma ja tyytyväinen hahmoonsa, sillä teksturoinnin jälkeen, jos haluaa hahmon ulkomuotoon tehdä muutoksia, muutoksia täytyy myös tehdä teksturointiin. Yleisissä ja yksinkertaisissa teksturoinnissa voidaan käyttää yhtä pääväriä ja teksturoida objekti 3ds Maxin sisällä. Tätä toimenpidettä käytämme napin kohdalla. Pikanäppäimellä M päästään materiaalieditoriin, jonka avulla voimme lisätä tekstuureita hahmoille ja objekteille.

Napin tapauksessa päädyttiin yksinkertaiseen turkoosin väriin ja samalla teksturoitiin myös napin läpi menevä naru mustaksi. Koska kyseessä on nappi, miellettiin se tässä

tapauksessa muoviksi ja haluttiin, että se heijastaa hieman valoa. Tämä onnistuu muuttamalla tekstuurin Specular-asetuksia. Asetuksilla voidaan vaikuttaa heijastuksen määrään, kiiltoon sekä heijastuksen väriin. Näitä asetuksia voi myös pehmentää ja 3ds Maxin sisällä pystyykin tekemään lähes kaikenlaisia tekstuureita, joita voi hahmojen tai muiden mallien kanssa käyttää. Materiaalieditorilla voi tehdä lähes kaikkea yksinkertaisista pinnoista läpinäkyviin pintoihin kuten lasiin sekä täysinheijastaviin peleihin tai mm. teräksen näköistä tekstuuria. Nämä kuitenkin eivät voodoo-nukelle sovi tähän tarkoitukseen ja oli mietittävä, miten tekstuurin voi kuvioittaa itse ja tehdä sille haluttuja jälkiä, kuten reikä silmässä sekä tikkejä.

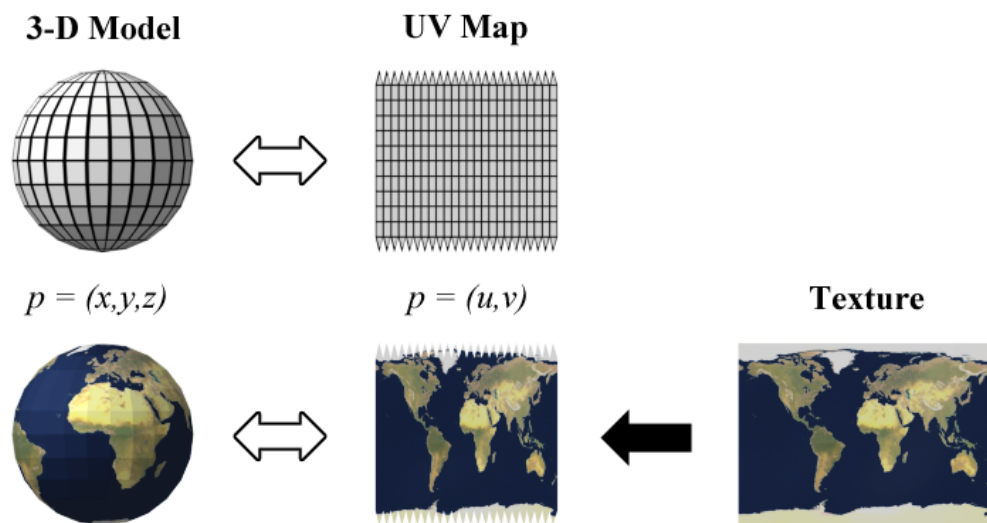
### 3.3.1 Bitmap

Koska yksinkertainen materiaali tähän tilanteeseen ei sopinut, täytyy luoda käsin tekstuuri, joka voitaisiin liittää hahmon skiniksi. Tässä tilanteessa vaihtoehdoksi jää luoda oma Bitmap, joka on digitaalinen kuva. Bitmap tässä tapauksessa liitetään tekstuurin tapaisesti hahmon pinnalle. Bit map toteutetaan etsimällä oikeanlaista tekstuurikuviota, joka sopisi hahmolle ja luodaan siitä oma tiedosto Photoshopilla. Tässä huomattiin, että vaikka .PSD, eli Photoshopin omat tiedostot, ovat tuettujen tiedostomuotojen listalla 3ds Maxissa, sitä ei saatu toimimaan oikein. Jostain tuntemattomasta syystä luotu tekstuuri heijasti valoa hahmon pinnalta, vaikka muita asetuksia ei ole säädetty. Tätä ei saatu toimimaan .PSD-loppuisella tiedostolla, vaan yritettiin yleistä vastaavaa, eli .PNG. Tällä tavalla tekstuuri saatiin näkymään halutulla tavalla.

Molemmat .PSD- ja .PNG-tiedostomuodot voivat sisältää layereita, kerroksia. Kerrokset voi helpoiten hahmottaa esimerkiksi piirtoheittimen kalvojen avulla. Ensimmäiselle kalvolle piirretään jotain ja se jätetään pinon alimmaiseksi. Toiselle kalvolle voidaan piirtää jotain vaikuttamatta alla olevaan kuvaan. Tämä auttaa varsinkin silloin, jos halutaan tehdä jotain päällekkäin ja että tämän jälkeen on vielä mahdollisuus uusaa, siirtää tai muuten muokata kuvaa. Valmiin Bitmapin/tekstuurin voi lisätä 3ds Maxissa pikanäppäimellä M, valitsemalla uuden tekstuurin - valitsemalla Maps - Diffuse Color - Bit map ja valitsemalla luotu tekstuuri. Hahmon osalla kuitenkin tuli määrittää, mikä osa 2D-kuvasta vastaa mitään kohtaa 3D-mallissa, joten tämän tekstuurin muodostamiseen käytetään UV mappia ja tämän prosessin tuloksena on ns. texturemap.

### 3.3.2 UV mapping ja unwrap

UV mapping tai UV-mappauksella tarkoitetaan 2D-kuvan heijastamista 3D-objektille. Nimellä UV viitataan akseliin 2D-tekstuurissa, jotka vastaavat x, y, z -akseleita 3D -mallissa [8]. UV-mappausprosessissa siis luodaan 2D-kuva, joka ohjelmiston avulla välitetään 3D-objektille, ja objekti saa skinin. Tämä 2D-kuva on nimeltään UV texturemap.



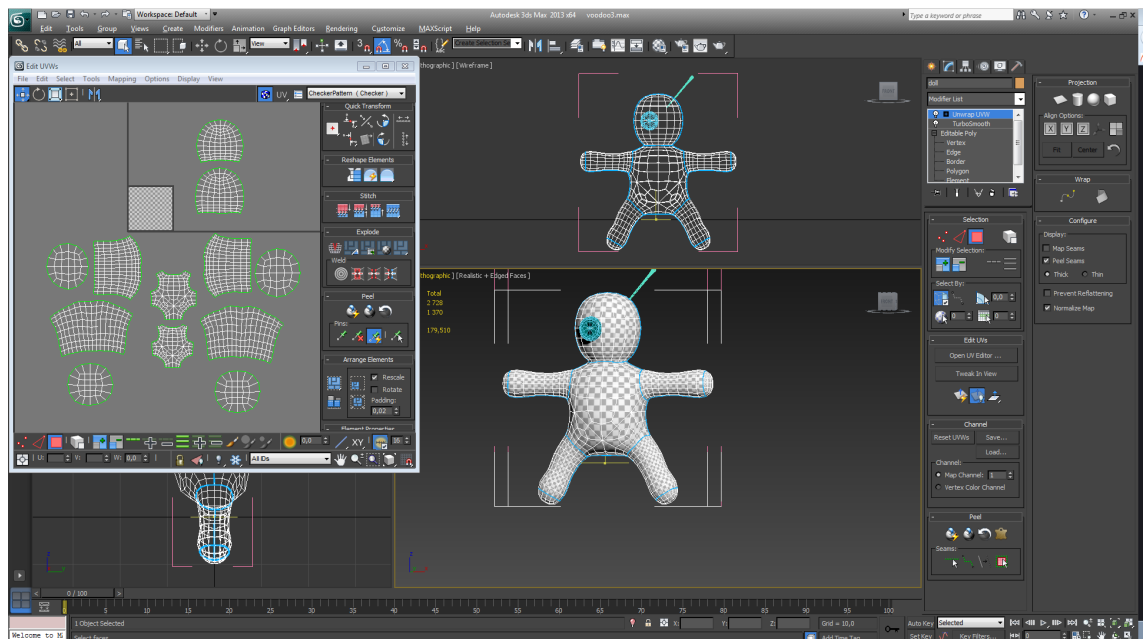
Kuva 10. UV mapping esimerkkikuva

Luodaksemme UV mapping -kuvan, 3D-malli jaetaan polygon-alueisiin, jotka levitetään ja tehdään 2D-kuvaksi. Tätä prosessia kutsutaan UVunwrappingiksi. Tätä toimenpidettä voidaan tehdä automaattisesti, käsin tai näiden yhteistyönä 3D-ohjelmiston sisällä. 3ds Maxin sisällä tätä toimenpidettä voidaan tehdä työkalulla Unwrap UVW, joka löytyy Modify-välilehden alta. Työkalun valittuaan valitaan Edit UVs -kohdan alta Open UV Editor. Tämän ikkunan sisältä löytyy kaikki tieto, jota UVwrappingiin käytetään. Tämä editori sisältää tiedot 3D-alueista ja sen avulla määritetään ja luodaan 2D-kuva, jota käytetään texturemappina.

Tämä vaihe on yksi teksturoinnin tärkeimmistä osista, sillä huono uvmap näyttää saumakohtista huonolta. Heti aloituksessa voidaan huomata, että saumakohdat näyttävät todella huonolta eikä automaattinen toteutustapa tähän toiminut laisinkaan. Tästä syystä täytyy alueet ja niiden saumakohdat tehdä itse. Tämä tapahtuu

valitsemalla Peel-kohdan alta Point-to-Point Seams, jolla voidaan rajata itse vertexien avulla rajat ja alueet. Nämä saumat tulee miettiä hyvin tarkkaan, sillä vaikka texturemap olisikin lähes täydellinen, on mahdollista, että saumat jäävät näkyviin. Tästä syystä saumoja tehdään sellaisiin paikkoihin, joista niitä vähiten näkee. Esimerkkinä on käden alapuoli, joka jää yleensä vartalon lähelle, eikä tästä syystä ole näkyvissä. Tässä rajauksessa kannattaa miettiä vaatteita ja niiden saumoja. Syyt ja ratkaisut ovat näissä yleisesti jo valmiiksi mietittynä ja niitä tarvitsee vain soveltaa käytännössä.

Saumojen teon jälkeen on aika luoda UVWrap. Vaiheessa valitaan jokainen alue erikseen ja alueen tyyppin mukaan käytetään Projection kohdan alta löytyviä painikkeita. Koska vartalo on suhteellisen tasainen, voidaan käyttää Planar Map tyyppiä muodostamaan 2D-kuva. Käsien kodalla voidaan käyttää Cylindrical Mappia parhaiden tuloksien saavuttamiseksi. Kun kaikki alueet on rajattu ja mapattu, on aika palata UV - editoriin ja erotella nämä rajatut alueet toisistaan. Alun alkaen kaikki ovat pienen neliön kohdalla päällekkäin, mutta ne tulee erotella ensiksi alla olevaan tapaan:

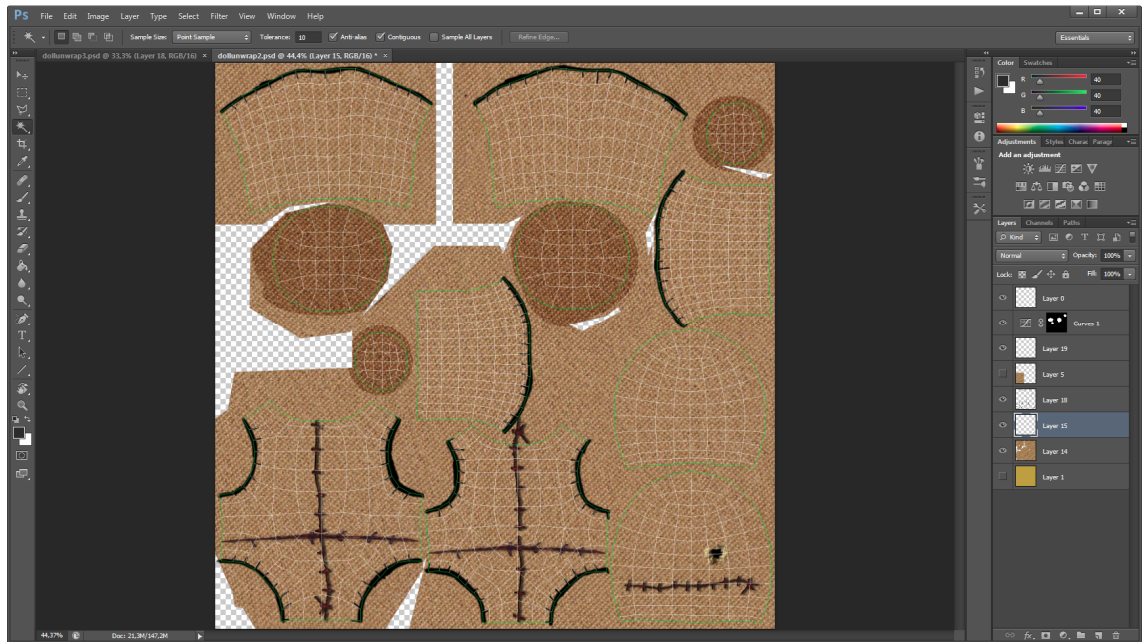


**Kuva 11. UVunwrap saumoista ja editorista**

Tämän jälkeen paloiteltuja alueita voi pienentää ja käännellä editorissa olevilla työkaluilla. Tavoitteena on siis saada ne takaisin kaikki pieneen laatikkoon keskellä mahdollisimman suurena, mutta kuitenkin siten, että ne eivät kosketa toisiaan eikä

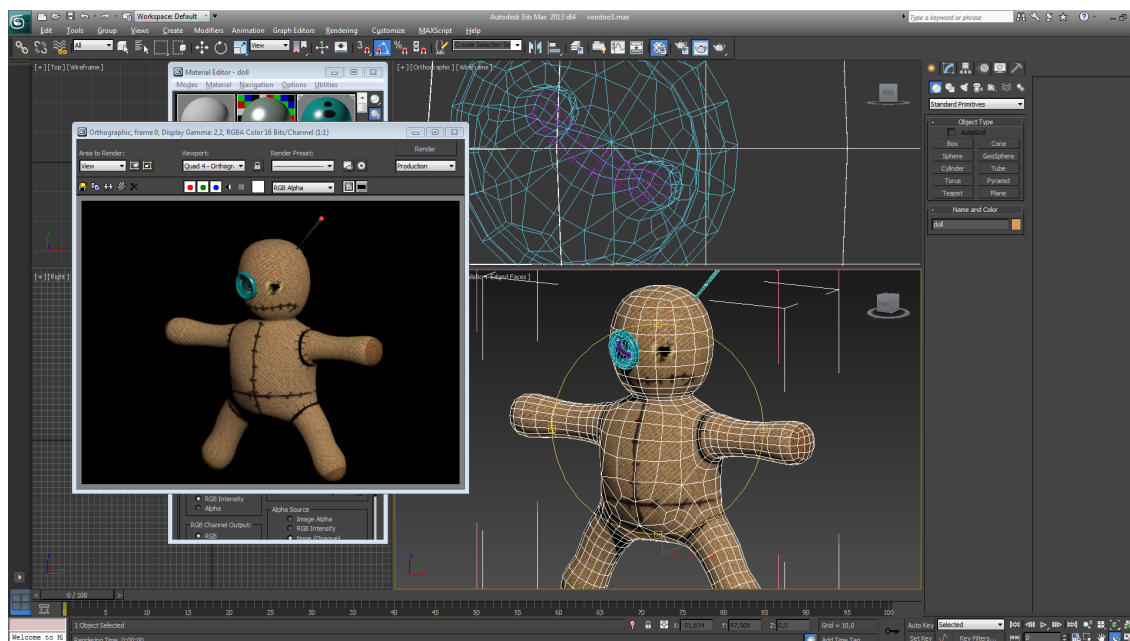


reunoja. Kun tulokseen ollaan tyytyväisiä, voidaan 2D-kuva renderöidä 3ds Maxista valitsemalla Tools - Render UVW Template. Tuloksena on siis yksikerroksinen kuva, jossa on tieto alueiden muodoista. Tämän jälkeen haluamaa tekstuuria voidaan alkaa laittamaan kuvaan. Kuvassa 12 voidaan huomata oikealla alhaalla olevien layereiden käyttötapoja, jossa Layer 0 sisältää ainoastaan tiedon aiemmin luoduista saumoista ja alueista. Tämä näkyy kuvassa vihreinä ja valkoisina viivoina.



**Kuva 12. UV texturemapin luominen**

Teksturoinnin valmistuttua tulee tekstuuri viedä takaisin 3ds Maxiin Bitmappina. Ennen tätä kuitenkin tulee piilottaa Layer 0, joka siis sisältää vihreät ja valkoiset rajat, koska tällöin skiniin ei myöskään tule näitä viivoja. Kun tekstuuri on valittu ja lisätty hahmolle, sitä voidaan milloin tahansa muokata, ja se päivittyy myös 3ds Maxiin sen tallennuksen yhteydessä. Bitmapin tekstuureihin lisäyksen jälkeen hahmo näytti alla olevan kuvan (Kuva 13) mukaiselta. Kokonainen texturemap on lisätty liitteeksi 1.



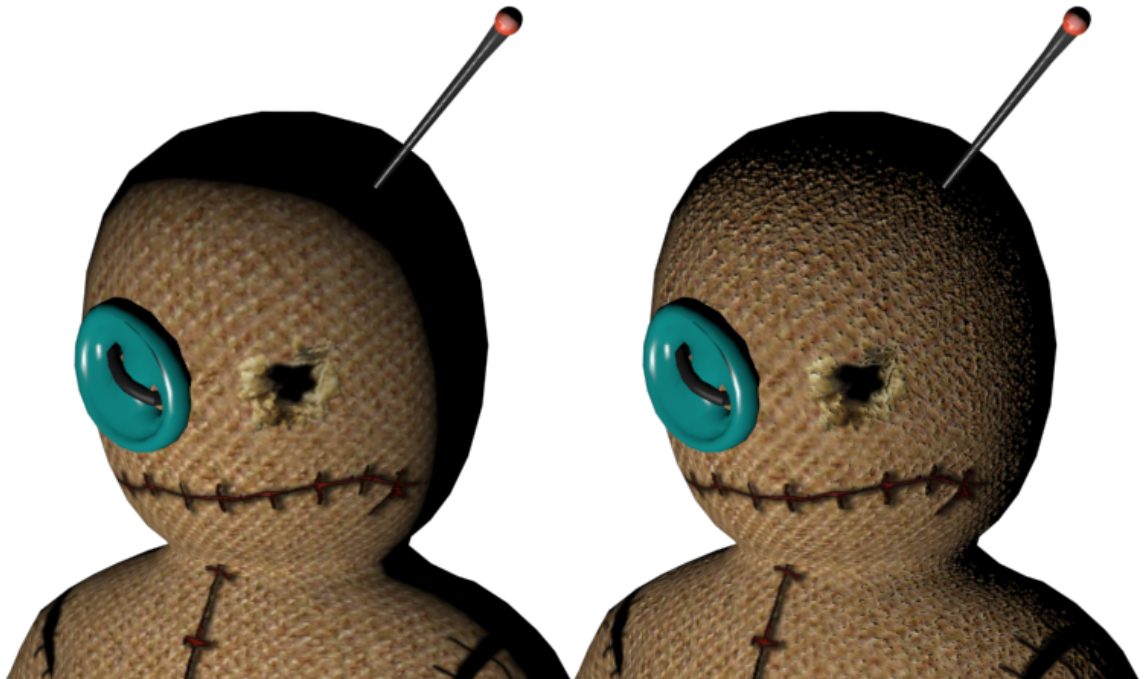
Kuva 13. Texturemapin lisäyksen jälkeinen tila

### 3.3.3 Bump map

Bump mapping on tekniikka, jolla voidaan simuloida epätasaisuuksia objektin pinnalla [12]. Käyttämällä bump mappia voidaan tehdä tasaisesta pinnasta kuviollinen halutulla tavalla. Kyseessä on siis grafiikan simulointitekniikka, joka käyttää hyväksi valon laskentaa ja muuttaa tekstuurin pintaa lisäämättä polygoneja. Bump-mapin käytöllä on hyvin vähän haittapuolia, vaikka sen lisäämät tulokset ovat ainutlaatuisia ja korvaamattomia. Bump mapin luonti aloitetaan Photoshopissa jo luodun texturemapin muutoksilla. Valitaan Image - Adjustements - Desaturate, joka muuttaa kuvan mustavalkoiseksi. Tämän jälkeen valitaan Filter - Other - High Pass, jolla voidaan tarkentaa kuvaa ja sen yksityiskohtia haluttu määrä. Itse käytin High Pass arvoa 3 tässä tapauksessa, sillä se näytti tuottavan tarpeeksi yksityiskohtia tekstuuriin. Tekstuurin tarkistuksessa tulee muistaa, että vaaleat kohdat bump mapissa ovat koholla tekstuurissa, ja tummat kohdat ovat painaumia.

Tekstuurin tallentamisen jälkeen siirrytään takaisin 3ds Maxin puoleen, jossa etsitään hahmolle tehty tekstuuri ja valitaan Maps - Bump ja lisätään tallennettu Bump map tähän kohtaan. Bump map ei näy lainkaan ennen renderöintiä, joten pikanäppäin renderöinnille, F9, auttaa Bump mapin määrittämisessä paljon. Bump mappia voi säätää arvosta 0 aina 999 arvoon saakka. Tässä tapauksessa käytettiin arvoa 40, jolloin

saatiin aikaan tarpeeksi yksityiskohtia kuitenkin liioittelematta. Kuva 14 kertoo visuaalisesti, kuinka vaikuttava bump map oikeasti on. Bump map lisätty liitteeksi 2.



Kuva 14. Bump map ennen ja jälkeen

### 3.4 Hahmon riggaus

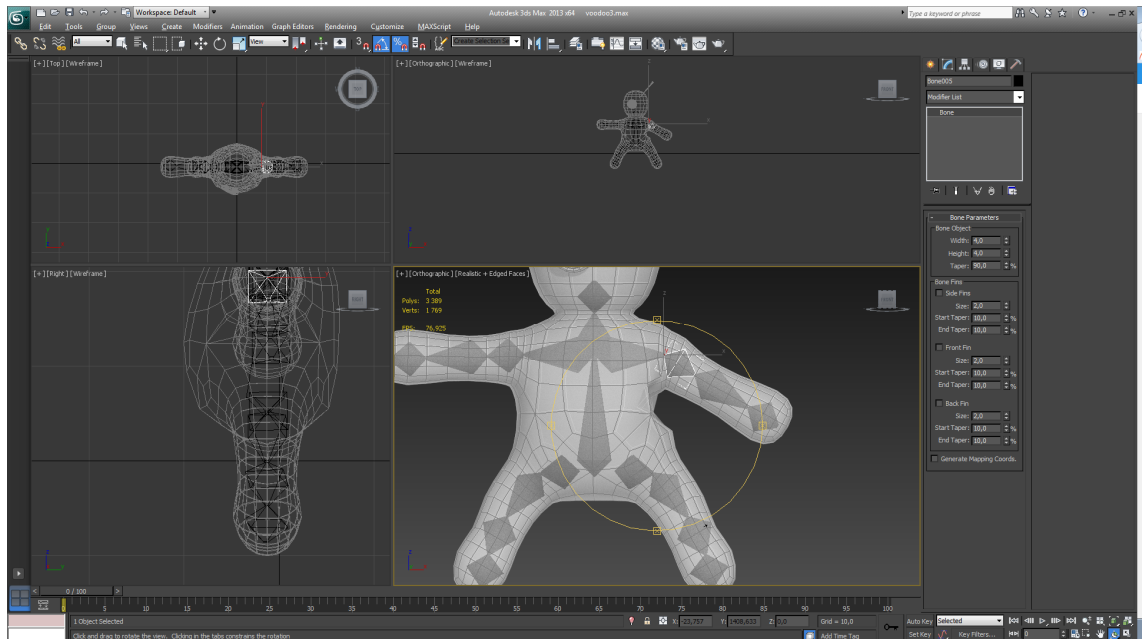
Hahmon riggauksella tarkoitetaan, että hahmolle lisätään luuranko tai luut, jotka valmistavat hahmon animointia varten. Riggausta varten on 3ds Maxissa valmis luuranko, jonka voi luoda valitsemalla Create - Systems - Biped. Tämä luuranko sisältää paljon luita ja on ihmismäisille hahmoille hyvä perusta. Kaikki tarpeellinen on luurangon osalta tehty valmiiksi, ja se kävisi myös tälle hahmolle loistavasti. Luotuja luita voi myös renderöidä ja niillä on useita asetuksia, jotka auttavat ja helpottavat varsinkin animointia. Fins, eli ns. lisäsiivekkeet ovat yksi apuosa, joita luihin voi lisätä. Nämä siivekkeet näyttävät selkeästi luun pyörimisen ja auttavat animoinnissa [10].

Koska haluttiin kuitenkin kokeilla uusia tapoja, rakennettiin luut itse erikseen. Tämä tapahtui valitsemalla Systems - Bones. Luurangon rakentaminen aloitetaan yleisesti lantion kohdalta ja sen jälkeen rakennetaan selkäranka. Luut rakentuvat samalla

tapaan kuin aiemmin todetut linkit. Ne ovat hierarkisia, mutta tässä tapauksessa hieman erilaisia. Child-luita ei voi kuitenkaan liikuttaa ilman parent-luuta, ne ainoastaan pyörivät niille määräytyvän pivot-pisteen ympärillä, joka on luulle määritetty keskipiste tai lähtöpiste. Luiden tekoon on Bone Tools -apuväline, joka löytyy valitsemalla Animation - Bone Tools. Työkalun avulla voidaan lisätä, yhdistää ja muokata luita sekä valita tarvittaessa uudelleen Parent-luu. Työkalulla voidaan myös kopioida toiselle puolelle tehdyt luut valitsemalla Mirror ja valitsemalla oikea akseli, johon luut peilataan. Jalkojen luiden lisäyksen kanssa ilmeni hieman ongelmia, sillä Connect Bonen avulla ohjelma vaihtoi suoraan Parent-luiksi uuden reisiluun, emmekä tätä halunneet. Tämän takia jouduin luomaan lantion luun, reisiluun ja jalan kaksi luuta erikseen ja tämän jälkeen linkkaamaan ne selkärankaan. Tämä ei ole suositeltava tapa, mutta tähän ratkaisuun muuta keinoa en löytänyt. Tämän takia lantiota ei voi myöskään pyörittää, koska sitä periaatteessa ei ole. Lantioluu on tässä tapauksessa siis linkattuna selkärankaan, ja tavallaan osana sitä. Luurankoa tehdessä on hyvä muistaa, että mitä enemmän luita tehdään, sitä monipuolisemmaksi ne voidaan määrittää ja sitä enemmän taipumista tiettyihin kohtiin saadaan aikaan. Mallintamisen aikana voidaan ottaa myös tämä huomioon ja tehdä luiden liitospisteisiin enemmän polygoneja, jolloin luiden ja niihin liitettävä skinin osa taipuu ja venyy paremmin. Jotta skinin ja luut saadaan liitettyä yhteen ja luuranko toimimaan animointia varten, tarvitaan skinnausta.

### 3.5 Hahmon skinnaus

Skinnaus eli luiden ja hahmon ihon tai meshin yhdistäminen on myös yksi tärkeimmistä vaiheista ennen animoinnin alkua. Vaiheen tarkoituksena on siis yhdistää hahmon skin luotuun luurankoon animointia varten. Tässä vaiheessa kestää yleisesti hyvin kauan ja prosessi on välillä turhauttava ja pitkäväteinen. Aloitetaan valitsemalla Modify - Modifier list - Skin. Tämän jälkeen valitaan Bones - Add, ja valitaan kaikki hahmoa varten aiemmin luodut luut. Nyt hahmoon on liitetty luodut luut ja itse skinnausprosessi voidaan aloittaa. Kuten kuva 15 osoittaa, pyörittämällä käsien luita, myös hahmon skin liikkuu, mutta tämä on luonut myös ongelmia. Hahmon kainalon alla oleva alue on määrätty olemaan osana olkaluun vaikutusaluetta, ja kääntämällä olkaluuta myös kylkiluiden alueella oleva skin kääntyy, eikä tämä ole tarkoitus.



**Kuva 15. Hahmon skinausprosessin alkuasetelmat**

Valitaan Parameters - Edit envelopes - Vertices. Nyt voimme vaikuttaa, paljonko luut vaikuttavat tiettyihin vertexeihin. Valitsemalla luun Bones-listasta, tässä tapauksessa olkaluun, ja valitsemalla Weight Properties -työkalun avulla vertexejä, voidaan niiden painotusta nostaa ja poistaa. Tämä skaala on 0 ... 1, jossa 0 tarkoittaa, että valitulla luulla ei ole valittuun verteksiin lainkaan painoarvoa, ja tällöin luun kääntelyllä ei ole vaikutusta vertexin sijaintiin. Tässä tapauksessa haluamme valita muuttuneet vertexit ja valita niiden painoarvoksi olkaluun kohdalla 0. Tämä toistetaan myös toiselle puolelle ja valitaan selkärangan luu ja asetetaan vertexin painoarvot 1. Luita voi myös käyttää siten, että luiden painoarvoa jaetaan useiden luiden välille, ja tämä on myös suositeltavaa. Myös tätä tekniikkaa voi mieltää oikeaan elämään ja miettiä, miten tietyt luut vaikuttavat ihon ja lihaksien liikkumiseen.

Muita ongelmia tuli myös vastaan, kun Skin modifier ei ollut valinnut ollenkaan yhtään luuta tiettyihin vertexeihin. Tämän pystyi toteamaan liikuttamalla koko luurankoa ja näkemällä, että tietyt vertexit pystyivät paikoillaan. Suurimpia ongelmia olivat kyljet sekä lonkat, jotka olivat kauimpana mistään luusta. Tämän ongelman olisi voinut mahdollisesta välttää käyttämällä luissa aiemmin mainittuja siivekkeitä. Kun säätelee ja muokkaa tarpeeksi kauan vertexien painoarvoja, päästään tulokseen, joka tyydyttää.

### 3.6 Hahmon animointi

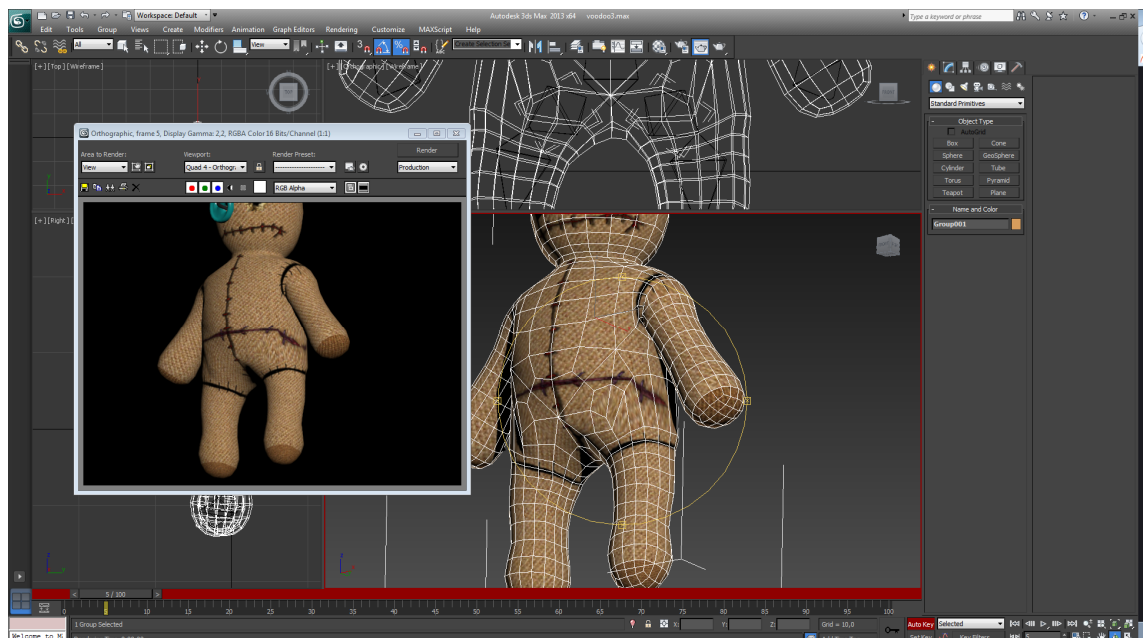
Animoinnin yksi päällisistä ongelmista on se, miten hahmolle halutaan toteuttaa animaatiot. Pyritäänkö toteuttamaan mahdollisimman realistiset animaatiot vai halutaanko korostaa tiettyjä osia esim. omalaatuista kävelytyyliä? Toinen ongelma on hahmojen raajojen mahdollisuus pyöriä ympäri, jolloin skin menee ns. rullalle tai kääntyy luonnottomaan asentoon. Tätä varten on luotu Inverse Kinematics, IK. Tällä työkalulla voidaan luoda luonnollisia liikeratoja mm. raajoille. Jos halutaan luoda liikerata esim. kädelle, tulee meidän määrittää alkupiste ja loppupiste IK:ta varten. Käden tapauksessa tämä olisi olkapää ja kämmen. Se toteutuksessa valitaan päätepiste, ns. child-objekti, ja valitaan Animation - IK Solver - HI Solver ja tämän jälkeen valitaan ns. parent-objekti, johon tämä liikerata päättyy, eli olkapää. Nyt kämmenen ja olkapään välille on syntynyt ketju, joka automaattisesti päivittyy ja animoituu, kun kämmentä liikuttaa. Lisätty IK solver laskee luiden liikkeen ja pyöriksen määrän sekä liikuttaa koko luuketjua child-objektin liikkeen mukaisesti (kuva 16). Kuvassa näkyy siis vertailu luun liikuttamisesta. Näitä hierarkisia ketjuja voi muodostaa useita ja niiden käyttö animointia varten on hyvin suositeltavaa.



**Kuva 16. Inverse Kinematicsin demonstrointia**

IK-ketjujen luonnin jälkeen voidaan aloittaa itse animointi. Tähän on useita tapoja, mutta esittelen tavan, jota itse käytin animaationi tekoon. 3ds Maxissa animointi onnistuu helposti ja suhteellisen vaivattomasti. Alhaalla on ns. timeline, eli aikajana,

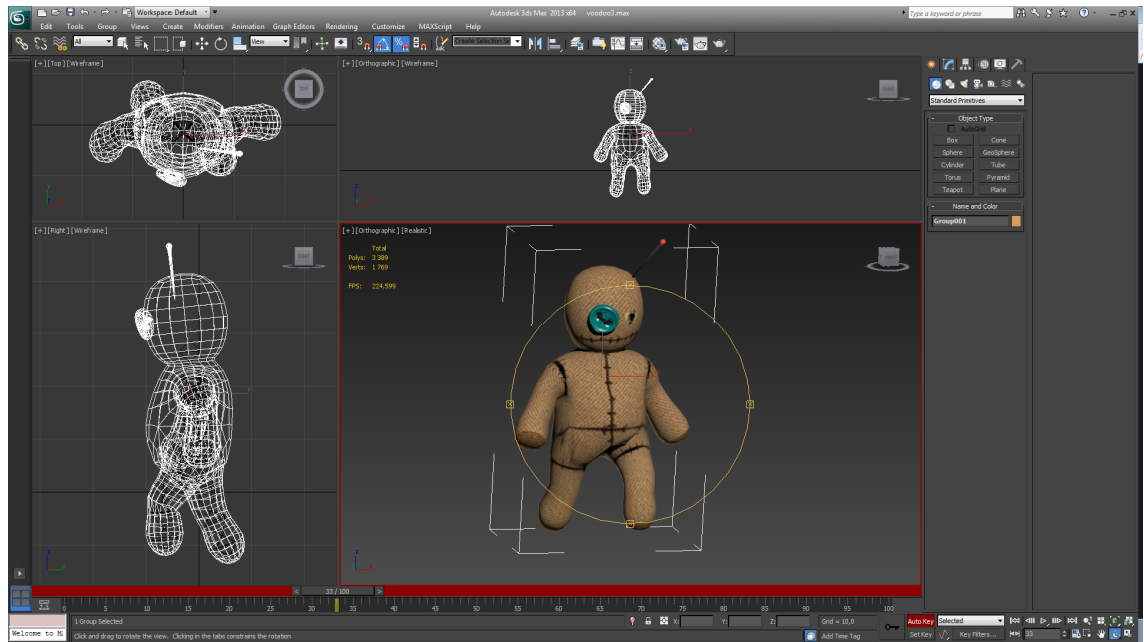
johon hahmon tietyt vaiheet tallennetaan keyframien avulla. Keyframe on siinä hetkiä ajassa, johon on tallennettu hahmon sillä hetkellä oleva asento. Animaatio aloitetaan keyframe 0 ja lopetetaan tässä tapauksessa keyframe 100. Hetkellä 0 hahmolla on perusasento, joka oli enemmän X-mallinen kuin suositeltu T-mallinen, joten alkuasentoa oli muutettava. Ennen tätä kuitenkin kopioitiin hetki 0 ja asetettiin se myös kohtaan keyframe 100. Tällä toimenpiteellä saadaan aikaan ns. animation loop, jolla pyritään luomaan asetelma, jossa animaatio ei näytä loppuvan, vaan se jatkuu saumattomasti koko ajan. Koska alku- ja loppuasento oli nyt tehty, voitiin näiden välille luoda itse liikkeitä. Tämän jälkeen valittiin Auto Key aikajanan alta, mikä asettaa keyframeja automaattisesti aina siinä aikajanan tilanteessa, jossa hahmoa liikutetaan. Tämän ero Set Key -toimintoon on se, että jos Auto Key ei ole päällä, ja hahmoa liikutetaan, sen liike tai sen hetkinen tilanne ei tallennu mihinkään tiettyyn hetkeen. Set Key -toiminnolla voidaan siis asettaa keyframe silloin, kun hahmon liikkeen tilan haluaa tallentaa tiettyyn hetkeen. Set Keyä yleisesti käytetään yksinkertaisissa animaatioissa ja sitä voisi myös käyttää tämän hahmon animoinnissa. Animoinnissa päädyttiin kuitenkin käyttämään Auto Keyä. Seuraavaksi valittiin keyframe 5 ja aseteltiin raajat luonnolliseen asentoon siten, että raajat roikkuvat alaspäin. Tämä toteutettiin luiden liikuttamisella, ja koska hahmo oli skinnattu skinnattu X-asentona, pieniä tekstuurin venymisiä löytyi.



Kuva 17. Hahmon lähtöasento animointia varten

Kuvassa 17 huomaa vasemman lonkan kohdalla venymistä tekstuurissa, ja tämän aiheutti lonkassa olevat vähäiset polygonit ja lähtöasento. Alkuasennon muokkaamisen jälkeen voitiin kävelyanimaatiota lähteä tekemään. Tässä vaiheessa päätettiin, että animoidaan viiden keyframen välein. Koska ensimmäisenä haluttiin tehdä luonteva kävelyanimaatio, animoitiin jalat ensimmäisenä. Keyframe 10:n kohdalla nostettiin vasen jalka ylös ja siirrettiin oikeaa jalkaa hieman taaksepäin. Koska käytettiin IK, polvet taipuivat luonnollisen oloisesti ilman muita toimia. Keyframe 15:ta kohdalla laskettiin vasen jalka takaisin alas ja vietiin oikeaa jalkaa vielä taaemmas. 20:n keyframen kohdalla vietiin vasen jalka taakse ja nostettiin oikeaa jalkaa ilmaan. Viimeiseksi 25 keyframen kohdalla vietiin oikea jalka edessä maahan ja vietiin vasempaa jalkaa taaemmas. Nyt ensimmäinen kävelykierto on valmis jalkojen osalta. Seuraavaksi animoitiin hahmon korkeus. Hahmo animoitiin siten, että kun jalkaa nostettiin, nostettiin myös itse hahmoa hieman ylöspäin ja laskettiin silloin, kun jalat ovat toisistaan kauimpana, eli hetki ennen jalan nostoa. Tämän loi luontevan liikkeen koko hahmolle kävelyssä. Tässä vaiheessa myös kallistettiin hahmoa hieman eteenpäin animaation ajaksi. Tämän jälkeen animoitiin kädet siten, että oikea käsi meni vasemman jalan tahtiin ja toisinpäin. Seuraavaksi animoitiin pää, mutta huomioon piti ottaa se, että pää yleensä animoidaan muutama keyframe jäljessä muiden raajojen suhteen. Tämä johtuu siitä, kun vartalo kulkeutuu alaspäin ja kun se lopulta lopettaa liikkeen, pää haluaa jatkaa matkaa vielä, jolloin se liikkuu myöhemmin. Tätä varten animoitiin pää 2 keyframea muun hahmon liikkeen jälkeen, eli 7,12 jne. Pään animoinnin jälkeen hahmon kävelyanimaatio oli lähes valmis. Pienen hienosäännön jälkeen kopioitiin koko kävelyanimaation keyframeet ja liitettiin ne toiseksi kävelykierroksi, jolloin voitiin tarkastella näiden liittymistä toisiinsa. Huomattiin, että animaatioiden välissä oli hieman omituisia liikkeitä johtuen ensimmäisestä keyframeesta, jossa siis hahmo seisoj paikallaan. Tämä keyframe poistettiin ja hahmon liike kävelykiertojen välissä luonnollistui.





**Kuva 18. Kävelyanimaation tekoa hahmolle**

Toinen animaatio, joka toteutettiin, oli hyppyanimaatio. Animaation alussa hahmo oli jälleen alkuasennossa, jonka jälkeen hyppyä varten ensimmäiseen keyframeen polvia ja käsiä taivutettiin sekä hahmoa kallistettiin ponnistusvaiheessa hieman eteenpäin. Tämän jälkeen jalat suoristettiin ja seuraavassa keyframessa nostettiin hahmoa ylöspäin ja siirrettiin hahmon käsiä lähemmäksi vartaloa liikkeen sulavuuden takaamiseksi. Hypyn yläasennossa myös levitettiin käsiä sivummalle ja hieman hahmon alaspäin tullessa levitettiin käsiä vielä hieman. Alastulossa jälleen taivutettiin polvia. Tämän jälkeen animoitiin pään asennot samaan tyyliin kuin kävelyanimaatiossa. Animaatioiden valmistumisten jälkeen on aika viedä hahmo Unityyn ja testata hahmon animaatiot.

## 4 Loppuesittely ja mietteet

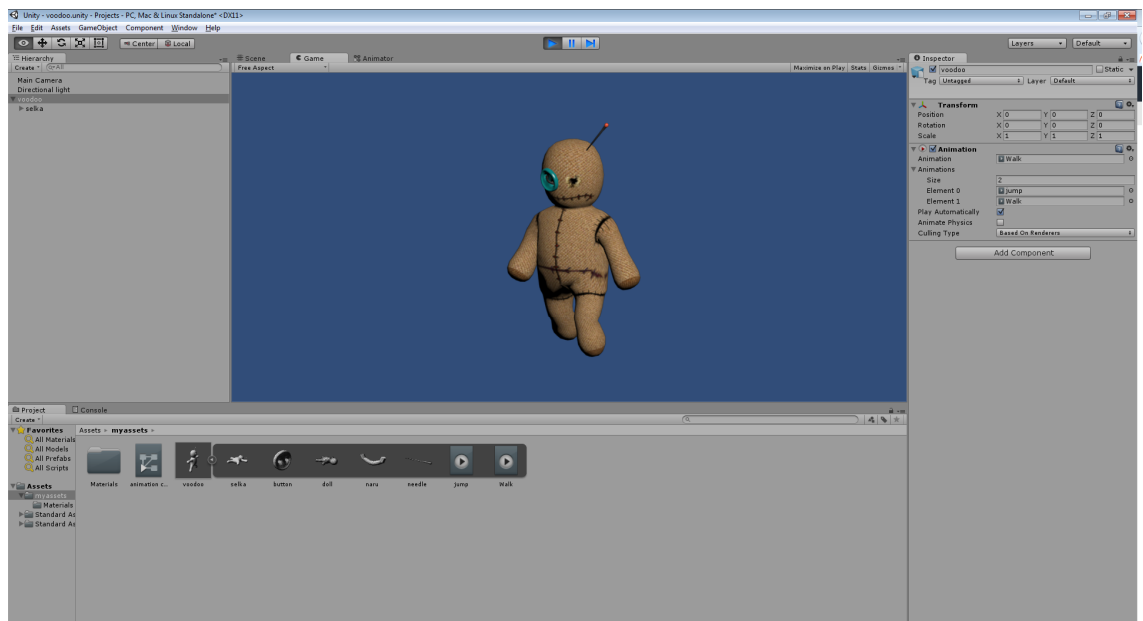
### 4.1 Testaus

Hahmon testaus päätettiin tehdä Unityllä. Unity mainostaa itseään ohjelmana, joka tarjoaa pelimoottorin lisäksi kaiken tarvittavan, jota tarvitaan, kun luodaan pelejä usealle alustalle. Työssä käytettävä versio on ilmainen, ja se sisältää kaikki työkalut pelien ja muiden sovellusten luontia varten. Tämän lisäksi se vaikutti helpolta käytettävältä, vaikka en sitä ei aiemmin ole kokeillutkaan. Yksinkertaiset käyttöliittymät sekä sisäänrakennettu manuaali auttoi sen ymmärtämistä paljon. Unityyn hahmon vientiin oli kaksi erilaista mahdollisuutta: luoda joko .FBX-tiedosto tai viedä 3ds Maxissa luotu tiedosto suoraan Unityyn. .FBX-tiedosto sisältää vähän tietoa, ja periaatteessa vain sen, mitä Unity tarvitsee. Toisaalta .MAX-tiedoston vieminen ohjelmaan on helppoa, mutta sen koko on suuri ja saattaa hidastuttaa käyttöä. Lisäksi se sisältää paljon turhaa tietoa, jota Unity ei voi käyttää. Tähän tarkoitukseen valittiin .FBX-vaihtoehto. Tiedoston luonti tapahtui File - Export, jonka jälkeen valittiin tiedoja, joita vietiin unityyn. Tässä oli mahdollisuus lisätä mm. valoja tai varjoja suoraan, mutta tätä emme tähän tarkoitukseen tarvinneet. Ohjelmasta ulosvienti tapahtui perusasetuksia käyttäen nopeasti ja vaivattomasti.

Tiedoston luonnin jälkeen Unityssä avattiin uusi projekti, jonka alle saattoi rakentaa oman ympäristönsä kansioita rakentamalla. Varsinkin suuremmissa ympäristöissä tästä on apua, mutta yhden hahmon testaukseen riitti yksi kansio. 3ds Maxista tuotu tiedosto vietiin luodun kansion alle, ja näin hahmo oli luotu Unityyn. Seuraavaksi valittiin hahmo, ja siirrettiin se Sceneen, yleiseen näkymään, jonka jälkeen hahmo näkyi ympäristössä. Huomattiin, että hahmoa ei näkynyt lainkaan, joten Model -valikon alta vaihdettiin Scale Factor - 1. Tämän jälkeen hahmo näkyi normaalisti, ja ympäristöön lisättiin myös Directional light valaisemaan hahmoa sekä uudelleen aseteltiin kamera. Nyt voitiin määrittää animaatiot ja niiden alku- ja loppuajankohdat. Nämä oli helppo tarkistaa 3ds Maxista, jossa ne määritettiin. Kävelyanimaatio alkoi framesta 5 ja loppui frameen 55. Hyppy animaatio alkoi framesta 65 ja päättyi framessa 95. Asetukset määritettiin Animation - Clips -kohdan alla ja valittiin valittiin Rig välilehden alta - Animation Type - Legacy. Tämän jälkeen palattiin Animations välilehdelle ja valittiin Add Loop Frame sekä Wrap Mode - Loop. Asetuksilla

varmistettiin, että liikkeet näyttävät hyvältä toistattaessa niitä useamman kerran peräkkäin.

Tämän jälkeen luotiin animaatioille Animation Controller valitsemalla Assets - Create - Animation Controller. Animation Controllerilla voidaan nimensä mukaisesti hallita ja toistaa useita eri animaatioita yksinkertaisesti. Animaatiot löytyivät valitsemalla hahmo ja valitsemalla Animator ja avaamalla Animation -kohdan, jossa näkyvät kaikki hahmolle määritetyt animaatiot. Näitä voitiin Play-napin avulla testaila ja kaikki näytti käyttäytyvän juuri kuin pitikin. Voitiin tässä vaiheessa todeta projektin onnistuneisuus.



Kuva 19. Hahmon animaation testaus Unity-ohjelmassa

## 4.2 Loppumietteet

Projektin tavoitteena oli luoda 3D-hahmo ja animoida se. Työssä suunniteltiin ja luotiin hahmon lisäksi myös tekstuurit ja kävely- sekä hyppy animaatiot hahmolle. Tavoitteisiin päästiin, ja työn lopputulos näyttää mielestäni hyvältä. Vaikka työ alun perin kuulosti suppealta, työhön kuului hyvin monta eri vaihetta ja menetelmää, jota työssä käytiin kattavasti läpi. Työssä esiteltiin vaiheet suunnittelusta mallintamiseen sekä riggauksesta ja skinnauksesta animointiin.

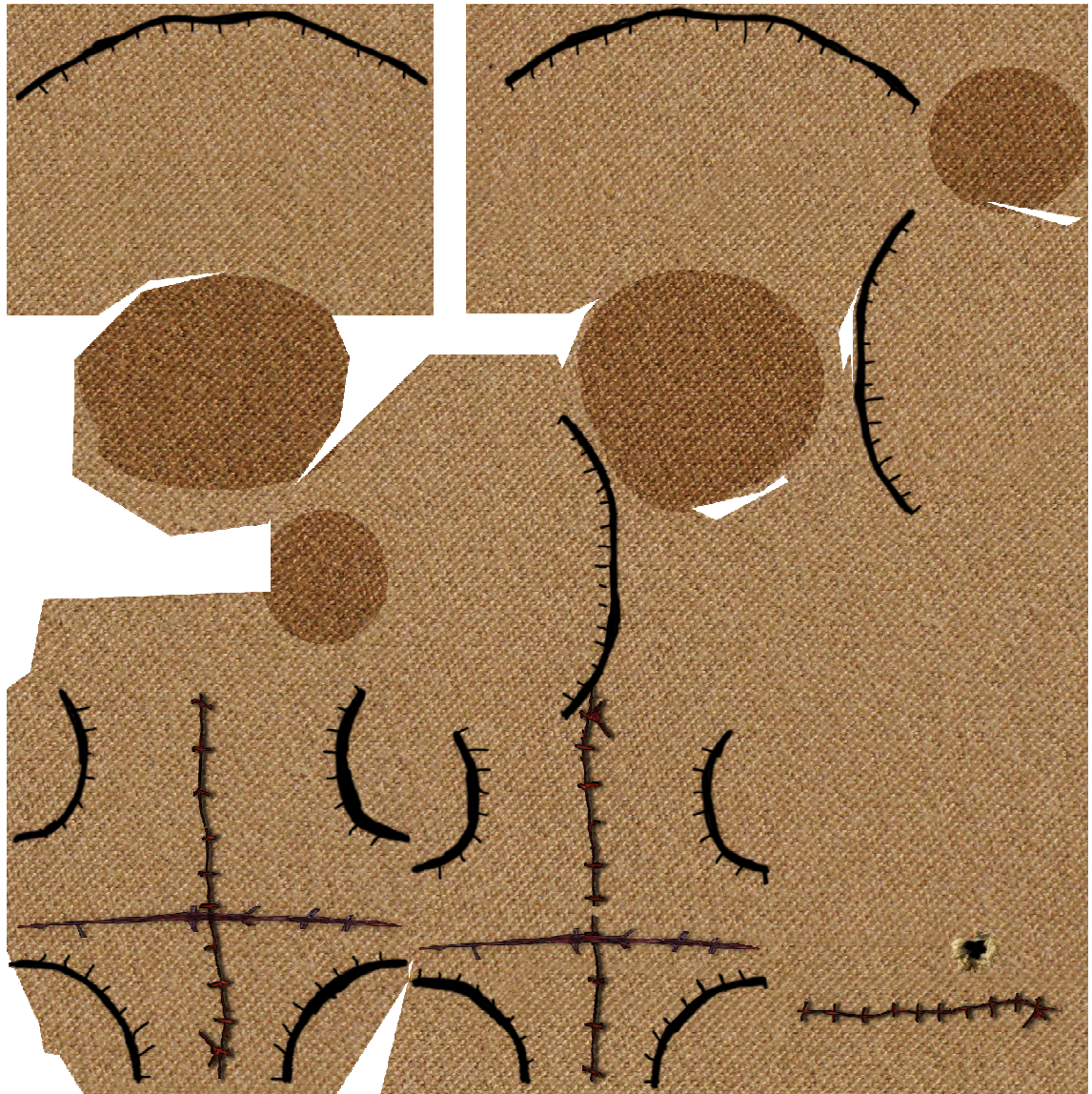
Tekniikat ja työtavat tulivat tutuiksi. Vaikka lopputulos olikin onnistunut, aloitettiin muutamaan kertaan projektin mallinnusosuus kokonaan uudelleen. Tämä siksi, että hahmon korjaamiseen menisi paljon enemmän aikaa kuin siihen, että luotaisiin se samaan vaiheeseen asti samassa ajassa. Tämä myös opetti parempia työtapoja ja nopeampia keinoja saavuttaa sama asia.

Vaikka eri vaiheisiin kuluikin oletettua enemmän aikaa, oli työ vaivan arvoista. Työssä käytetyt ohjelmistot olivat kattavia ja monipuolisia. Ne antoivat paljon uutta opittavaa tulevaisuutta varten. Oppimieni asioiden perusteella voin mielestäni hyvin mallintaa vaikeampia ja suurempia 3D-malleja sekä animoida niitä. Tämä ei tietenkään tarkoita sitä, että parhaat työtavat tai tekniikat on löydetty, vaan kokemuksen kautta löytyy aina jotain parannettavaa.

## Lähteet

- 1 David C & Co. 2014. 3D modeling. Verkkodokumentti.  
<[http://en.wikipedia.org/wiki/3D\\_modeling](http://en.wikipedia.org/wiki/3D_modeling)> Luettu 4.1.2015.
- 2 Introduction to 3D Modeling . 2014. Verkkodokumentti.  
<<http://www.animationarena.com/introduction-to-3d-modeling.html>> Luettu 4.1.2015.
- 3 Mikko Paananen, Alfred Degan. 2014. Animaatio. Verkkodokumentti.  
<<http://fi.wikipedia.org/wiki/Animaatio>> Luettu 4.1.2015.
- 4 Masters, Mark. 2014. 3ds Max, Maya LT or Blender - Which 3D Software Should I Choose for Asset Creation? Verkkodokumentti.  
<<http://blog.digitaltutors.com/3ds-max-maya-lt-blender-3d-software-choose-asset-creation/>> Luettu 5.1.2015.
- 5 Blind pose. 2014. Verkkodokumentti. <[http://www.ssbwiki.com/Blind\\_pose](http://www.ssbwiki.com/Blind_pose)> Luettu 5.1.2015.
- 6 Linking and Unlinking Objects. 2015. Verkkodokumentti.  
<<http://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/3DSMax/files/GUID-4115DE9B-D636-4A42-904E-71259A843E0A-htm.html>> Luettu 5.1.2015.
- 7 McMahon, Drongle. 2012. What does UV Mapping do exactly? Verkkodokumentti. <<http://community.secondlife.com/t5/Mesh/What-does-UV-Mapping-do-exactly/td-p/1710229>> Luettu 6.1.2015.
- 8 Rick Farmbrough. 2014. Verkkodokumentti.  
<[http://en.wikipedia.org/wiki/UV\\_mapping](http://en.wikipedia.org/wiki/UV_mapping)> Luettu 6.1.2015.
- 9 Raymond Grier, Eduard Pintilie. 2014. Verkkodokumentti.  
<[http://en.wikipedia.org/wiki/UV\\_mapping](http://en.wikipedia.org/wiki/UV_mapping)> Luettu 6.1.2015.
- 10 Bones System. 2014. Verkkodokumentti.  
<<http://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/3DSMax/files/GUID-E6164716-CFA9-4DE9-9976-F8A58850461F-htm.html>> Luettu 6.1.2015.
- 11 Inverse Kinematics (IK). 2015. Verkkodokumentti.  
<<http://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/3DSMax/files/GUID-516E301F-E911-429F-9337-9FA7FAD49BB6-htm.html>> Luettu 6.1.2015.
- 12 Cory Janssen. Bump Mapping. Verkkodokumentti.  
<<http://www.techopedia.com/definition/1961/bump-mapping>> Luettu 6.1.2015.

Texture map



### Bump map

