



TAMPEREEN  
AMMATTIKORKEAKOULU

# 3D-OHJELMA FYSIOTERAPEUTTISEEN KÄYTTÖÖN

SKELETOR 2016

Pia Lampinen

Opinnäytetyö  
Marraskuu 2016  
Tietojenkäsittely  
Pelituotanto



## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietojenkäsittely  
Pelituotanto

LAMPINEN, PIA:  
3D-ohjelma fysioterapeuttiseen käyttöön  
Skeleton 2016  
Opinnäytetyö 29 sivua  
Marraskuu 2016

---

Opinnäytetyön tavoitteena oli syventää osaamista 3D-mallintamiseen, saada kokemusta Unity-pelimoottorin käytöstä sekä oppia ohjelmoimaan C#-kielellä. Opinnäytetyön tavoite oli myös tuottaa toimiva 3D-ohjelma fysioterapeuttiseen käyttöön toimeksiantajan ideasta. Painopiste opinnäytetyössä oli käytännön osuudella. Toimeksiantaja on jäsenkorjaaja Hyvinkäältä, Vesa Hietanoro. Vaatimuksena työlle oli saada toimeksiantajalle toimintakelpoinen ohjelma, jota hän pystyy hyödyntämään omassa työssään.

Opinnäytetyössä käydään läpi osa-alueet, joita ohjelman toteuttaminen vaatii ja toimintamekaniikkojen toteutuksen ratkaisut. Työssä on pohdintaa projektin onnistumisesta ja siitä miten opinnäytetyön tuotosta voi hyödyntää jatkossa. Pohdinnassa käyn myös läpi, mitä kehityskohteita ohjelmassa vielä on sekä miten ohjelmasta voisi saada rahaa tuottavan tuotteen markkinoille. Työ täytti sille asetetut vaatimukset ja ohjelmaa voidaan ladata tulevaisuudessa internetistä omaan käyttöön.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Business Information Systems  
Game Development

LAMPINEN, PIA:  
3D Program for Physiotherapeutic Use  
Skeleton 2016  
Bachelor's thesis 29 pages  
November 2016

---

The purpose of this thesis was to deepen the author's knowledge of 3D modeling and to get experience of using the Unity game engine, as well as to learn to program in the C# language. An additional objective of the thesis was to produce a functional 3D program for physiotherapeutic use on basis of a development idea suggested by the commissioner of this thesis. The main focus of the thesis was the practical part. The commissioner is a physiotherapist from Hyvinkää, Mr. Vesa Hietanoro. The requirement was to make a functional 3D program that the commissioner could utilise in his own work.

The thesis goes through areas of what the program requires to be fully operational. The work is a reflection on the success of the project and how it can be utilized in the future. The thesis also deals with the shortcomings of the program and how to monetize it in the future. The project met all the requirements set for it and will be available for downloading from the Internet for personal use in the future.

---

Key words: unity, 3D-modeling, programming

## SISÄLLYS

1	JOHDANTO.....	7
2	UNITY 5.....	9
3	KONSEPTI.....	11
4	SUUNNITTELU .....	12
	4.1 Käyttöliittymä .....	12
	4.2 Syöte ja kontrollit: miksi päädyin näihin ratkaisuihin? .....	13
	4.3 Ohjelman optimointi .....	13
	4.4 Grafiikkojen optimointi .....	14
5	TOTEUTUS .....	16
	5.1 Grafiikat .....	16
	5.2 3D-mallintaminen .....	17
	5.3 Tukipisteet .....	19
	5.4 Mekaniikka luiden kääntelyn takana ja Gimbalin lukko .....	20
	5.5 Luun kierteen mukaan korostevärien vaihtaminen.....	21
	5.6 Liukusäätimien ja kameran liikuttelun toimintaperiaate.....	22
	5.7 Tiedostotoiminnot ja ponnahdusikkunat.....	24
6	POHDINTA.....	27
	LÄHTEET.....	29

## ERITYISSANASTO

Asset(s)	Valmis elementti Unityssa, esimerkiksi 3D-malli tai käyttöliittymäpalkki/nappula.
Bake	Baking, eli ”beikkaaminen” on tapa, jolla saadaan luotua korkeapolygonisen 3D-mallin yksityiskohdat matalapolygoniseen 3D-malliin tekstuurin avulla. Useimmissa 3D-mallinnusohjelmissa, kuten Blenderissä, voidaan beikata tekstuuri.
.csv	Tiedostomuoto, jota voidaan lukea muun muassa Excelillä tai ohjelmointiympäristöillä. Lyhenne tulee sanoista ”comma-separated values” ja sillä käytännössä tallennetaan tekstitiedostoon yksinkertaista taulukkomuotoista tietoa.
.fbx	Autodeskin omistama ”Filmbox”-tiedostomuoto. FBX-tiedostomuotoa käytetään antamaan yhteensopivuus digitaalisten sisältöä luovien ohjelmien välille. Useimmista 3D-mallinnusohjelmista pystytään nykyään tuomaan ulos malleja .fbx-tiedostomuodossa. Useimmat pelimoottorit, kuten Unity ja Unreal Engine, tukevat .fbx-tiedostomuodoissa tuotuja malleja.
Full Screen	Tietokoneella käytettävän ohjelman ruudun tila. Full Screen tarkoittaa koko näyttöä, jolloin ohjelma on reunattomasti koko näytöllä, eikä taustalla auki olevia ohjelmia voi aktivoida klikkaamalla Full Screenissa olevan ohjelmaruudun ohi.
Object	3D-malli voi sisältää monta pienempää palasta, jotka yhdessä muodostavat 3D-mallin.
Pivot Point	Tukipiste, joka voidaan määrittää 3D-malleille haluttuun paikkaan. Sen ympäri objektin halutaan kääntyvän sitä kääntäessä.

Polygoni	Polygoni, suljettu kuvio, joka muodostuu vähintään kolmesta sivusta. Sivujen välissä on polygonin pinta eli surface. Nämä yhdessä muodostavat objektille pinnan, kun objektille halutaan lisätä esimerkiksi tiedot tekstuurista, geometriasta tai valaistuksesta. Kokonaisuudesta saadaan 3D-malli.
Unityn Asset Store	Unityn Asset Store on virtuaalinen kauppa. Sieltä voi ostaa esimerkiksi valmiita, jonkun toisen tekemiä koodipätkiä tai graafisia elementtejä käytettäväksi Unity-projekteihin.
Workflow	Workflow tarkoittaa työnkulkua ja työssä edettävää työjärjestystä.

## 1 JOHDANTO

Opinnäytetyöni aiheena on tehdä 3D-ohjelma fysioterapeuttiseen käyttöön. 3D-mallintaminen on minulle entuudestaan tuttua, mutta opinnäytetyössä on haasteena Unityn käyttö ja ohjelmointi C#-kielellä. Haluan edistää ja inspiroida ihmisten terveyden hoitamista sekä parantamista nykyteknologian avulla. Painopiste opinnäytetyössä on selkeästi käytännön työn puolella, sillä kokonaisen ohjelman rakentamiseen vaaditaan paljon aikaa ja uuden opettelua.

Toimeksiantajana 3D-ohjelmalle on hyvinkääläinen yrittäjä, Vesa Hietanoro, jonka alkuperäinen idea ohjelman luonti on. Hietanoro on jäsenkorjaaja ja hän ollut yrittäjänä alalla useita vuosia. Toteutan ohjelman pääasiallisesti omien taitojeni pohjalta noin puolessa vuodessa. Tavoitteena on tehdä toimiva 3D-ohjelma, jolla pystytään tarkastelemaan ihmisluurangon yli 200 luuta. Pyrin saamaan ohjelman julkaisukelpoiseksi, jotta muut aiheesta kiinnostuneet pystyvät lataamaan sen internetistä omaan käyttöönsä.

Opiskelen ohjelmaa varten itselleni täysin uuden ohjelmointikielen, Unityn, dokumentaation ja muiden internetistä löytyvien lähteiden avulla. Käytän toimeksiantajan ammattitaitoa 3D-mallintaakseni tarvittavan ihmisluurangon ohjelmaa varten. Hyödynnän koulussa oppimiani 3D-mallinnustaitojani, Unityn käytön perusteita sekä ohjelmointiosaamistani.

Valmistun tietojenkäsittelijäksi ja olen erikoistunut pelituotantoon, ja opinnäytetyössäni kohtaavat kaikki tulevaisuuden pelialan töissä tarvittavat taidot. Opinnäytetyö auttaa syventämään Unityn osaamistani ja jatkossa voin myös ohjelmoida C#-kielellä vaivattomammin. 3D-ohjelman suunnittelu ja toteutus ovat hieno lisä ansioluettelooni sekä portfoliooni, mikä pelialalla on koulutusta ja tutkintoa merkittävämpää.

Oppinäytetyön pääasiallisena tietolähteenä toimii internet, sillä tiedon uusiutuminen ja ohjelmien päivittyminen tapahtuvat todella nopeasti. Tietoa, jota tarvitsen ohjelman tekemiseen, löytyy hyvin rajallisesti kirjallisista lähteistä.

Opinnäytetyö ei sisällä liitteitä, mutta Skeletor 2016 -ohjelma tulee olemaan vapaasti ladattavissa internetistä.



## 2 UNITY 5

Unity-kehitysympäristö mahdollistaa 2D- ja 3D-pelien, sovellusten sekä kokemusten luomisen. Unity-kehitysympäristö sisältää muun muassa Unity-pelimoottorin ja -editorin.

Unitylla on internetissä laaja yhteisöverkosto ja foorumi, josta löytyy tukea ja vinkkejä apua kaipaaville. Unitylla on myös kattava dokumentaatiokirjasto, josta löytyy tietoa Unityn omista ohjelmointimeteodeista tarkemmin. Myös Unityn suosion takia monet yrityksen ulkopuoliset tahot sekä ohjelman kehittäjät ja harrastajat ovat tehneet paljon ohjetutoriaaleja, miten voi aloittaa lähes minkä tahansa projektin tekeminen Unitylla. (Unity Documentation 2016.)

Unity 5 tukee useita alustoja tietokoneista mobiililaitteisiin. Luettelo Unityn tukemista alustoista on kattava (Unity Multiplatform 2016):

- iOS, Android, Windows Phone, Tizen
- Windows, Windows Store Apps
- Mac
- Linux/Steam OS
- WebGL
- Playstation 4, Playstation Vita
- Xbox One, Xbox 360
- Wii U
- Nintendo 3DS
- Oculus Rift, Google Cardboard, Steam VR, Playstation VR, Gear VR, Microsoft Hololens
- Android TV, Samsung SMART TV, tvOS

Unitysta on tällä hetkellä saatavilla neljä erilaista versiota käyttäjän tarpeiden mukaan. Unityn Personal-versio on ilmainen, mutta se sisältää pakollisen Unityn oman käynnistysruudun, kun sillä tehtyä sovellusta avataan. Personal-versiossa on suppeammin ominaisuuksia verrattuna maksullisiin Unityn versioihin, mutta ominaisuusvalikoima ei ole kuitenkaan liian suppea, vaan se sopii juuri pieniin projekteihin sekä harrastuskäyttöön.

Muut versiot (Plus, Pro ja Enterprise) maksavat kuukaudessa 35 dollarista ylöspäin ja ne on tarkoitettu isompiin yritys- ja ammattilaisprojekteihin. Maksullisissa versioissa on etuna 20-40 prosentin alennukset Unityn omassa Asset Storessa. (Unity Store 2016.)

Unityn ohjelmointikielien ovat pääasiassa C# ja UnityScript. C#-ohjelmointikieli muistuttaa paljon Javaa ja UnityScript puolestaan JavaScriptia. Unityssa voi myös ohjelmoida Boo-ohjelmointikielillä, joka muistuttaa Pythonia. Kaikille Unitysta kiinnostuneille ohjelmoijille löytynee suhteellisen tuttu kieli, joten käyttäjän ei tarvitse opetella pelkästään Unityn takia täysin uutta kieltä. C#-ohjelmointikieli on tutkitusti yksi nopeimmista kielistä, sillä C#:lla toteutetut metodit, komennot, ynnä muut ovat selkeämmin määritellyjä verrattuna muiden kielten vastaaviin metodeihin, joten sovellusta ajettaessa resursseja ei mene hukkaan. Edellä mainitun vuoksi C#:lla ohjelmointia suositellaan lämpimästi Unitylla työskenneltäessä. UnityScript on aloittelijaystävällisempi ja sitä suositellaankin monella Unityn foorumilla juuri heille, mutta jos kokemusta on jo aiemmin kertynyt, niin uutta Unity-projektia aloittaessa kannattaa ehdottomasti työskennellä C#:lla. (Codementor 2016.)

### 3 KONSEPTI

Skeletor 2016 -ohjelman alkuperäinen määritelmä on 3D-ohjelma fysioterapeuttiseen käyttöön. Kyseessä on 3D-ohjelma, josta on karsittu kaikki ylimääräinen toiminnallisuus ja jossa liikutellaan ihmisen luurangon luita visuaalisesti. Jokaista luuta pystyy liikuttamaan ihmiselle luonnollisten asteiden mukaan ja myös kuvakulmaa voi kääntää käyttäjän tarpeiden mukaan. Skeletor 2016:n on tarkoitus helpottaa ihmisen tukirangan visuaalista esittämistä esimerkiksi fysioterapeutin asiakkaan nikamaongelmista, ilman että fysioterapeutin tarvitsee osata käyttää 3D-mallinnusohjelmaa. Skeletor 2016 on hyvä työkalu fysioterapeutille ja myös asiakkaalle edullisempi ratkaisu kuin röntgen tai magneettikuvaus.

Työn edetessä Skeletor 2016:lle on löytynyt muitakin hyviä käyttötarkoituksia, kuten opetusvälineenä luiden nimien opiskeluun tai pohjana digitaaliselle grafiikalle, kun halutaan tuottaa mahdollisimman realistinen ihmisen poseeraus. Ohjelman jakaminen muille halukkaille käyttäjille tulee tapahtumaan internetin välityksellä.

Täysin vastaavia ilmaisohjelmia ei ole markkinoilla. Internetistä löytyy pari erilaista selainversiota, esimerkiksi Anatronica ja Kineman. Anatronican huono puoli on muun muassa, että se on tehty Unitylla ja Chrome-selain ei tue enää Unity Web Player-lisäosaa (plugin) ilman haastavaa kikkailua. (Unity Blog 2015.) Mozillalla saa Anatronican ihmisenäkymän auki, mutta ennen kuin mitään edes pääse testaamaan, ohjelma kysyy, haluaako käyttäjä ostaa normaalin vai ammattilaisversion. Käyttökokemuksesta en voi arvioida. Anatronica näyttää kuitenkin sisältävän luiden lisäksi myös muita kerroksia, kuten elimistön ja lihakset. Anatronican grafiikka näyttää nopealla silmäyksellä laadukkaalta ja todenmukaiselta. Kineman myös tarjoaa maksullista versiota, jossa pääsee liikuttelemaan ihmisen kehon kaikkia luita. Maksuttomassa versiossa on rajallinen määrä liikuteltavia luita, lähinnä käden ja jalan isoimpia luita. Luiden liikuttelu tapahtuu liikusäätimillä, jotka on sijoitettu hankalasti ja nimetty epäselvästi. Kinemanin luurangon luut on nimetty ja mallinnettu todenmukaisesti, myös kääntöasteet vaikuttavat oikeilta. Mitään projektin tallennusmahdollisuuksia ilmaisversiossa ei ole.

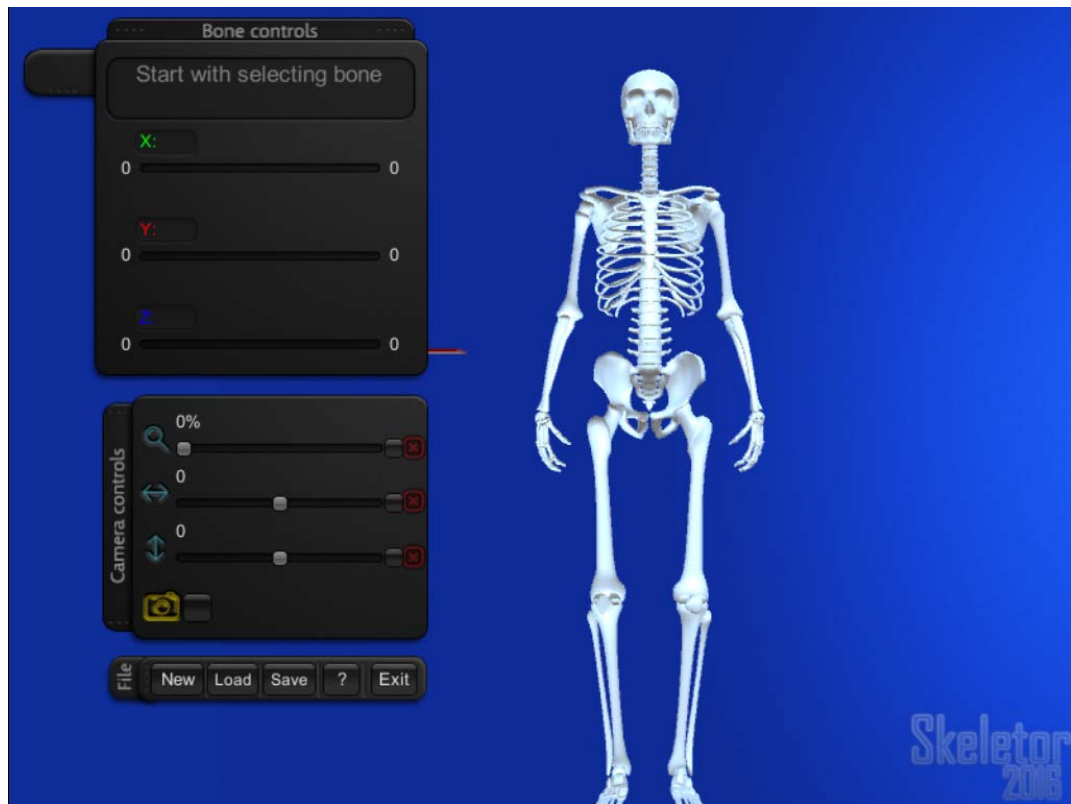
## 4 SUUNNITTELU

Oli ohjelma mikä tahansa, niin sen suunnitteluun kannattaa varata aikaa; hyvin suunniteltu on puoliksi tehty. Skeletor 2016 -ohjelman suunnittelu tapahtui yhdessä asiakkaan kanssa, sillä alkuperäinen idea oli asiakkaalta peräisin. Asiakas toivoi, että opettaisin hänelle Blender 3D -mallinnusohjelman käytön, mutta sain paremman idean. Ehdotin, että tekisin hänelle opinnäytetyönäni oman ohjelman, jossa on vain ominaisuudet, joita hän tarvitsee. Tästä alkoi ohjelman suunnittelu.

### 4.1 Käyttöliittymä

Ohjelman käyttöliittymä on pyritty tekemään mahdollisimman yksinkertaiseksi käyttäjäkuntaa ajatellen. Käyttöliittymässä ei ole liikaa nappuloita, vaan se on hyvin yksinkertainen ja nappuloissa on yksinkertaiset symbolit.

Käyttöliittymässä on kolme eri 'palkkia' (Kuva 1). Ensimmäisessä palkissa lukee aina valitun luun nimi ja se on tarkoitettu kyseisen luun liikutteluun. Toisessa palkissa on kaikki kameran kontrollit eli mistä kuvakulmasta valittua luuta halutaan tarkastella. Kamera pyörii vaakatasossa 360 astetta luurangon ympäri, mutta alle 180 astetta pystysuuntaan. Kamerakontrollipalkki sisältää myös kuvankaappaus-työkalun, jolla käyttäjä voi halutessaan ottaa kuvan ohjelman ruudun sisällä. Kuva tallentuu automaattisesti ennalta määrättyyn kansioon, josta käyttäjä voi halutessaan esimerkiksi jatkokäyttää sitä digitaalisen piirustuksen pohjana tai lähettää sen eteenpäin. Tiedoston asetuksia varten on kolmas palkki, mikä on hyvin tyypillinen ratkaisu kaikissa kuvankäsittely- sekä 3D-mallinnusohjelmissa. Kolmas palkki sisältää toiminnot: uusi projekti, tallenna, avaa, lisää tietoa/apua ja sulje ohjelma.



KUVA 1. Skeletor 2016 käyttöliittymä

#### 4.2 Syöte ja kontrollit: miksi päädyin näihin ratkaisuihin?

Kaikki syötteet ja kontrollit pyrin toteuttamaan siten, että käyttäjän ei tarvitse katsoa tutoriaalivideoita oppiakseen, miten ohjelmaa käytetään. Monia 3D-mallinnusohjelmia käytettäessä joutuu katsomaan useita tunteja opetusvideoita ennen kuin voi edes aloittaa työskentelyn ja tämänkin jälkeen pitää edelleen ongelman tullessa kohdalle etsiä vastauksia internetin syövereistä minuutteja, ehkä jopa tunteja. Selitän tarkemmin toteutusotsikon alla luiden käännön liukusäätimiä ja kameran liikuttelun toimintaperiaatetta sekä muita valikoita.

#### 4.3 Ohjelman optimointi

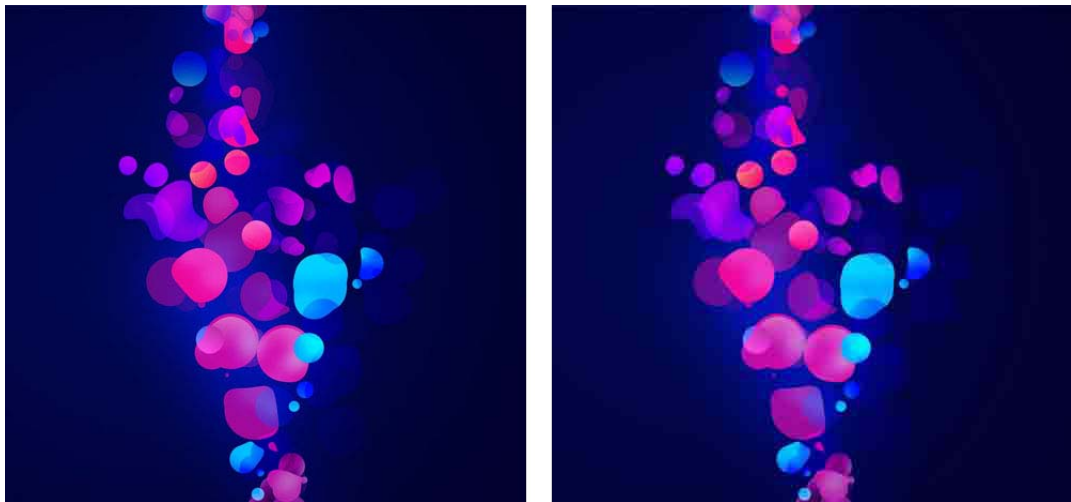
Unitylla pelien optimoinnista löytyy paljon informatiivista tietoa, varsinkin mobiilipelejä varten. Skeletor 2016 on kooltaan pieni ohjelma, eikä se vie paljoa tietokoneen resursseja. Ohjelman käyttöalusta on pc, joten Skeletorin optimointiin ei ole ollut tarvetta käyttää enempää aikaa ja tutkimista. Jatkossa vastaavia projekteja tehtäessä on mieles-

täni tärkeää kiinnittää huomiota enemmän optimointiin, etteivät ohjelmat kasva tarpeettoman suuriksi sillä turhat säikeet koodissa syövät koneen prosessointitehoa.

#### 4.4 Grafiikkojen optimointi

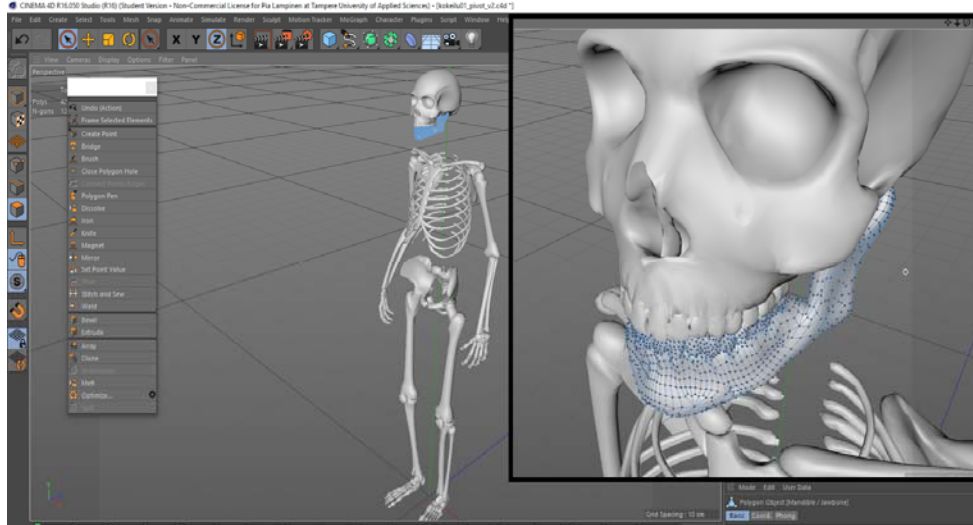
Ennen Skeletor 2016 -projektia oli optimointi grafiikkojen osalta entuudestaan tuttua. 3D-mallinnusvaiheessa on hyvä jättää turhat polygonit pois käyttäjäkokemusta piläämatta. Tarpeettoman suuret tekstuurit 3D-malleissa syövät turhaan tehoja, ja tässä ohjelmassa tekstuurit eivät vaikuta juuri käyttökokemukseen, joten niiden määrä on lähes olematon.

Kuvassa 2 näkyy tarkemmin katsottaessa, kuinka rakeinen oikeanpuoleinen kuva on. Mitä tarkemmalta kuva näyttää, sitä raskaampi kuva on. 3D-malleissa tekstuurit ovat 2D-kuvia. Eli 2D-tekstuurikin vaikuttaa siihen kuinka raskas, tai tarkemmin sanottuna muistia syövä, malli on ohjelmassa pyörittää. Vasemman puoleisen, alkuperäisen kuvan, koko on 43 kilobittiä ja oikeanpuoleinen, pakattu kuva, on 6 kilobittiä. Ohjelmaa toteutettaessa kannattaa miettiä kuinka läheltä 3D-mallia tarkastellaan: esimerkiksi jos mallin merkitys ei pelissä ole suuri ja se näkyy vain kaukaa, ei tekstuuriin kannata silloin tuhlaa suuria määriä resursseja aivan turhaan.



KUVA 2. Kuvanlaatua havainnollistava ero erikokoisissa kuvissa

Skeletor 2016 -ohjelmassa on vain yksi 3D-malli: luuranko itse, joten oli varaa tehdä tarkempaa työtä polygoneilla. Kuvassa 3 näkyvät vaaleansiniset pisteet ovat verteksejä (vertice). 3-4 verteksiä muodostaa tason (face).



KUVA 3. Luurangon leukaluun verteksit

Mitä enemmän 3D-mallissa on vertekseistä muodostuvia tasojia, sitä raskaampi 3D-malli on. Useissa peleissä, joissa on 3D-malleja, käytetään matalapolygonisia-malleja, joiden päälle huijataan korkeapolygoninen-malli käyttämällä hyödyksi 3D- tai tekstuuriohjelman ”bake” – ominaisuutta. Käyttämällä bake – ominaisuutta, eli ”beikkaamalla”, saadaan yksityiskohtainen kuva korkeapolygonisesta mallista tekstuurikuvaan. Tällä menetelmällä saadaan edullisesti hienon näköistä jälkeä peleihin, joissa on 3D-malleja.

Kuvassa 4 on jalkajousi matalapolygonisena mallina Blender 3D-mallinnusohjelmassa (vasen) ja viereisessä kuvassa sama jalkajousi huijattuna tekstuurien avulla korkeapolygoniseksi malliksi Substance Painter 2 -teksturointiohjelmassa.



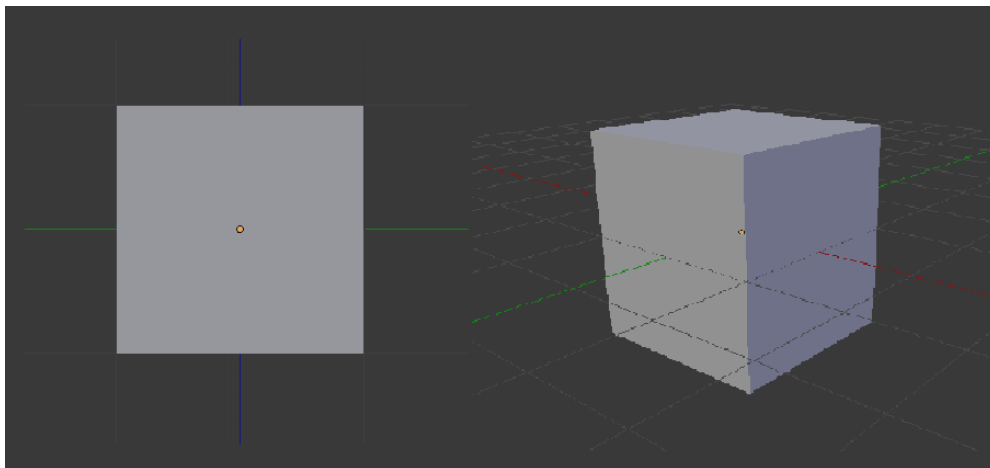
KUVA 4. Matalapolygoninen 3D-malli ja leivottu 3D-malli

## 5 TOTEUTUS

Skeletor 2016 toteutettiin Unity 5 -ohjelmalla. 3D-mallintamiseen käytin Cinema 4D-mallinnusohjelmaa. Tutustuin projektin yhteydessä myös Unreal Engineen, mutta Unity vastasi enemmän käyttötarpeita ja oli entuudestaan tuttu ohjelma. Ohjelmointikielenä käytin C#-kieltä, joka vaikutti henkilökohtaisesti helpoimmalta opetella.

### 5.1 Grafiikat

Ohjelman käyttötarkoitus on olla helpotettu ja karsittu 3D-ohjelma täsmäkäyttöön. 3D-kuviin on käytännössä tuotu yksi ulottuvuus lisää 2D-kuvaan verrattuna (katso kuva 5). Tämä mahdollistaa kuvan tarkastelun mistä tahansa kulmasta, mikä on hyvä mahdollisuus, kun halutaan tarkastella ihmisluurankoa kattavasti mahdollisimman pienillä kustannuksilla.



KUVA 5. 2D- ja 3D-kuvien ero havainnollistettuna.

Grafiikat ovat näkyvimmissä osassa ohjelmia, kuten tässäkin projektissa. Peruskäyttäjä ei yleensä ajattele sen tarkemmin taustalla toimivia mekaniikoita, vaan keskittyy olennaiseen: siihen mitä näkee. Erittäin tärkeän 3D-mallin grafiikoista Skeletor 2016:ssa tekee se, että sillä pyritään havainnollistamaan ranka- tai asentovikoja jäsenkorjaajan asiakkaille. On todella tärkeää, että nikamat ja luut on aseteltu oikein ja oikeassa kulmassa sekä luut on mallinnettu mahdollisimman täydellisen muotoiseksi. Kaikki luu-



rangot tietenkin ovat omanlaisiaan ja oman mallisiaan, ei ole kahta samanlaista, mutta Skeletor 2016 -ohjelmaan pyrin löytämään kultaisen keskitien mallin toteuttamiseen. Tulevaisuudessa projektiin voisi lisätä useamman “oletusluuranko-mallin”: miehen, naisen, lyhyen, pitkän, vanhan, nuoren ja niin edelleen.

Työvälinepalkkien taustaväriksi valitsin yksinkertaisen mustan rakeisella tekstuurilla tuomaan mielenkiintoisuutta palkkiin. Musta väri pitää palkit tarpeeksi esillä, sekä valkoinen teksti luo todella erottuvan kontrastin teksteihin, jotka käyttäjän on tarpeellista huomata. Ohjelmaa käynnistettäessä näkyy Skeletor 2016 -teksti, joka toimii myös ohjelman logona. Sama teksti näkyy myös vesileimana oikeassa alanurkassa itse ohjelmassa; esimerkiksi jos otetaan kuvankaappauksia ohjelmasta, näkyy mitä ohjelmaa on käytetty. Logon käyttö muistuttaa tekijänoikeuksista. Yleisesti väriteema on sininen, sillä mielestäni sininen väri ilmaisee kliinistä ja luotettavaa, kuten esimerkiksi sairaalaympäristöissä yleensä käytetään.

## **5.2 3D-mallintaminen**

Lähes puolet ohjelman toteutuksesta oli 3D-mallintamista, arviolta noin neljäkymmentä prosenttia projektiin käytetystä ajasta meni mallintamiseen. Malli on toteutettu Cinema 4D -ohjelmalla, joka oli entuudestaan tuttu. Cinema 4D ei ole mielestäni paras mahdollinen valinta 3D-mallinnusohjelmaksi, mutta aikaa ei mennyt turhaan oikean työvälineen etsimiseen. Luita ihmisellä on yli 200 ja jokaisen merkitys on todella tärkeä, joten mallinnusosuus piti tehdä huolellisesti ja ajan kanssa. Kaikki luut tehtiin liikuteltaviksi, osa luista liikkuu samassa ryhmässä resurssien säästämisen vuoksi, esimerkiksi kämmenessä olevat luut laitettiin liikkumaan yhdessä (kuva 6).



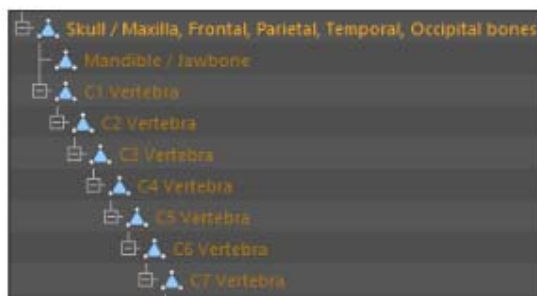
KUVA 6. Esimerkki kämmenessä olevasta luuryhmästä, luun nimessä on mainittu, että kyseessä on ryhmä.

Luurangon mallintamisen aloitin ylhäältä alas sekä vasemmalta oikealle. On hyvin tärkeää miettiä työjärjestys etukäteen, jotta mallintamisen aikana ei mene aikaa hukkaan pohtiessa, mitä on jo tehnyt ja mitä pitäisi vielä tehdä. Osan luista pystyi “peilaamaan” (eli vasemman käden luut oikean puolimmaisiksi käden luiksi), joten se säästi runsaasti aikaa mallintamisosuudessa.

Kuvasta 7 näkee, että luuranko sisältää pitkän listan luita, joista jokainen on mallinnettu erikseen omaksi kohteeksi (object). Mallinnusvaiheessa oli jo tärkeää nimetä luut oikein, sillä toteutin myöhemmin koodinpätkän, joka lukee suoraan kohteen nimen ja tulostaa sen käyttäjälle kertoen, mikä luu on valittuna. Hierarkian toteutus täsmälliseksi on myös hyvin tärkeää, sillä hierarkiassa ylempänä oleva luu hallitsee alempana olevaa. Kuvassa 8 näkyy, kuinka C1 Vertebra on ylempänä luuna muille Vertebra-luille, mikä tarkoittaa sitä, että jos C1 liikkuu, niin listassa alemmat liikkuvat sen mukana.



KUVA 7. Esimerkki hierarkian merkittävästä osuudesta.



KUVA 8. Ote hierarkiasta Cinema 4D:ssä

### 5.3 Tukipisteet

Kun kaikki listan luut oli mallinnettu, asetin niille tukipisteet (pivot point). Tukipiste on näkymätön piste, jonka ympäri kohde (object) liikkuu. Tässä tapauksessa jokaisen kohteen, luun, luonnollinen tukipiste on siellä, missä kyseistä luuta liikuttava nivel on.

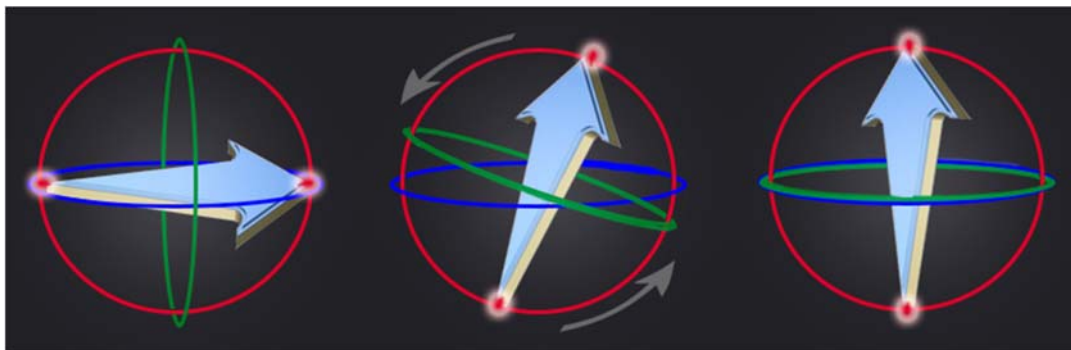
Tukipisteen asettaminen tehdään 3D-mallinnusohjelmassa, ja Unity tunnistaa sen automaattisesti, kun malli viedään ohjelmaan esimerkiksi .fbx-tiedostomuodossa. Unityn

puolella voidaan valita, halutaanko käyttää mallin omaa tukipistettä vai ohjelman yleistä tukipistettä.

#### 5.4 Mekaniikka luiden kääntelyn takana ja Gimbalin lukko

Luita käännellään koodissa rotate euler angles -komennolla, jota käyttäjä komentaa visuaalisesti liukusäätimillä. Rotate on suomeksi käännä, ja se komentaa haluttua asiaa kääntymään. Euler angles, eli Eulerin kulmat, ovat Leonhard Eulerin määritelmä kiinteän kolmiulotteisen kappaleen suunnista. Kyseisiä parametreja käytetään, kun halutaan kertoa, missä asennossa kappale on kolmiulotteisessa tilassa (euklidinen avaruus). Jokainen luu saa kolmiulotteisen asentonsa kolmen eri parametrin avulla: ensimmäinen kierto  $x$  kulmalla  $\alpha$ , toinen kierto  $y$  kulmalla  $\beta$ , ja viimeinen kierto  $z$  kulmalla  $\gamma$ . (MathWorld, Euler Angles.)

Skeletor 2016 -ohjelmassa kyseisen komennon käyttäminen toimii juuri niin kuin pitääkin, sillä jokaisella luulla on omat maksimi ja minimi rajoitteensa. Muissa projekteissa kannattaa kuitenkin harkita kyseisen komennon käyttämistä, sillä se saattaa aiheuttaa lukon kohteen suunnan muuttamiseen. Ilmiötä kutsutaan nimellä Gimbal lock, suorasuomenno voisi olla ”nivelliitoksen lukko”, käytän tekstissä kuitenkin termiä Gimbalin lukko. Gimbalin lukko tapahtuu, kun menetetään yksi kolmiulotteisista kääntöasteista (Kuva 11).



KUVA 11. Havainnollistava kuvasarja Gimbalin lukon syntymisestä.

Gimbalin lukko on tila, jossa yksi liikesuunnista menettää liikumisasteensa joutuessaan samansuuntaisesti toisen pyörimisakselin kanssa. Edellä mainittu ilmiö degeneroi kolmiulotteisen tilan kaksiulotteiseksi. Sana ’lukko’ on hieman harhaan johtava, sillä

yksikään pyörimisakseli ei varsinaisesti lukitu. Kaikki kolme pyörimisakselia ovat yhä vapaita liikkumaan oman akselinsa ympäri, vaikka yksi pyörimisakseli on menetetty.

Kuvassa 11 tapahtuu Gimbalin lukko -tila, kun punaisella havainnollistettua kääntöastetta muutetaan siten, että sinisen ja vihreän kääntöasteen arvot muuttuvat identtiseksi. Huomioin kyseisen ilmiön luiden kääntöasteisiin vaikuttavaa koodia rakentaessa, mutta totesin, että Skeletor 2016 -ohjelman nykyisessä versiossa ilmiö ei tule olemaan ongelma, sillä kaikilla luilla on x, y ja z -kierroille tiukat rajoitteet. (Euler (gimbal lock) Explained, Youtube 2009.)

### 5.5 Luun kierteen mukaan korosteväriin vaihtaminen

Asiakkaan erityistoiveena ohjelmalle oli, että se kertoisi käyttäjälle, kuinka kierteellä joku luu on verrattuna luun oletustilaan.

Toteutin ohjelmaan toiminnon, joka näyttää käyttäjälle, kuinka kierteellä luu on, lisäämällä siihen punaista väriä (katso kuva 12). Mitä enemmän luussa on kierrettä alkupe-  
räiseen asentoon verrattuna, sitä punaisempi luu on.

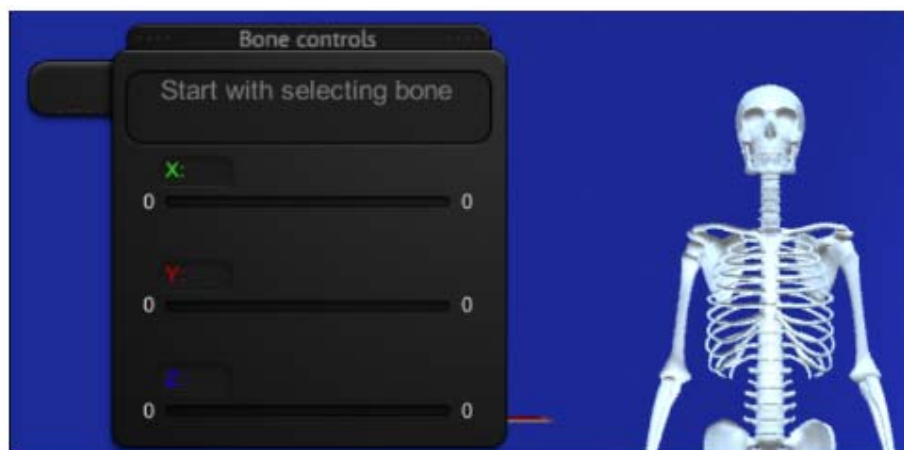


KUVA 12. Havainnollistava kuvasarja reisiluun kierteen lisäämisen vaikutuksesta luun väriin.

Punainen väri saadaan luun korosteväriksi asteittain, joten käyttäjän on helppo havainnollistaa luiden virheasentoja sekä tarkastella luurangon kokonaisuutta visuaalisesti.

## 5.6 Liukusäätimien ja kameran liikuttelun toimintaperiaate

Luun liikuttaminen tapahtuu valitsemalla ensin luu, ja tästä ylin palkki ilmoittaa tekstilä “Start with selecting bone” (Kuva 13). Kun luu on valittu osoittimella 3D-luurangosta, ylimmässä palkissa lukee valitun luun latinankielinen nimi ja 3D-mallissa valittu luu näkyy käyttäjälle vihreänä korosteväriin ansiosta. Tämän jälkeen käyttäjä pystyy vaihtamaan ylimmän palkin X-, Y- ja Z-arvoja liukusäätimillä, jotka ovat luun kääntöasteita eteen-taakse, oikealle-vasemmalle ja ylös-alas.



KUVA 13. Ohjelman käytön aloitus, luita ei ole vielä valittu

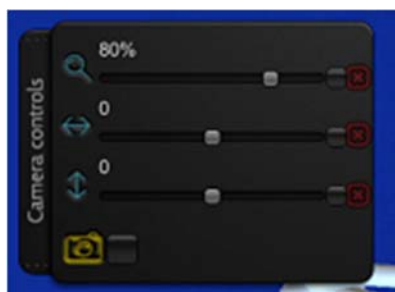
X-, Y- ja Z-kirjasimen värit ovat: punainen, sininen ja vihreä, jotta käyttäjät muistavat paremmin esimerkiksi mihin vihreän asteen muuttaminen vaikuttaa. Jos luuta ei voi kääntää Z-asteella, silloin liukusäätimen arvoa ei voi muuttaa, vaan se näkyy oletusarvoisesti nollana (katso kuva 14).



KUVA 14. Ruudunkaappauskuvan ottamishetkellä leukaluu (Mandible Jawbone) on aktiivinen ja valittuna.

Ylimmässä työskentelypalkissa luun nimen vasemmalla puolella on Reset-nappi. Kyseistä nappia painamalla voidaan nollata aktiiviseen luuhun tehdyt muutokset ja näin palauttaa luu oletusasentoonsa. Reset-nappi näkyy kuvassa 14.

Kameran liikuttelu tapahtuu ohjelmassa myös liukusäätimillä. Kameran liikutteluun vaikuttava työskentelypalkki näkyy kuvassa 15. Jokaisella kameran liikutteluun tarkoitettulla liukusäätimellä on oma nollausnappinsa, mikä parantaa käyttömukavuutta, kun käyttäjä haluaa palauttaa kameran lähtötilanteeseen. Nollausnapit on sijoitettu liukusäätimen oikealle puolelle, jokainen nollausnappi vaikuttaa vain viereiseen liukusäätimeen. Luurankoa pystyy tarkastelemaan lähemmin ylimmällä liukusäätimellä. Edellä mainitun liukusäätimen arvon kasvaessa tiettyihin pisteisiin kamera pyrkii keskittämään itsensä aina viimeisimpään aktiiviseen luuhun. Toinen liukusäädin vaikuttaa kameran horisontaaliseen sijaintiin: vaaka-asennossa kameraa saa pyöritettyä 360 astetta luurangon ympäri, eli 180 astetta vasempaan ja 180 oikealle. Kolmas liukusäädin vaikuttaa vertikaaliseen kameraan: kameraa voi liikuttaa ylös ja alas pienessä kääntökulmassa, jotta yksittäistä luuta saa tarkasteltua paremmin. Täysin ylä- tai alapuolelle kameraa ei saa, sillä se on ohjelman kannalta tarpeetonta.



KUVA 15. Kameran säätämiseen vaikuttava työskentelypalkki.

Toteutin ohjelmaan myös sisäisen ruudunkuvakaappaustoiminnon. Tämä lisää tietoturvaa kokemattomalle käyttäjälle, sillä kuvakaappaus otetaan ainoastaan Skeletor 2016 -ikkunasta. Näin käyttäjä ei virheellisesti vahingossa ota kuvaa taustalla olevista ohjelmista, työpöydän kuvakkeista tai yksityisistä tiedoista. Kuvakaappausnappia painaessa ohjelma ilmoittaa käyttäjälle “Screen capture saved!”, jonka jälkeen kuva löytyy paikallisesta Skeletor 2016 –kansioista (katso kuva 16).



KUVA 16. Oletuskansio kuvakaappauksille.

Kuville tulee oletusarvoinen nimi “Screenshot” ja numero kuvankaappaukselle. Numero kasvaa sitä mukaan, mitä enemmän käyttäjä on ottanut kuvia. Käyttäjä voi kuitenkin jälkikäteen muuttaa nimiä haluamikseen. Kuvat tallentuvat .png-tiedostomuodossa.

## 5.7 Tiedostotoiminnot ja ponnahtusikkunat

Kolmas palkki sisältää perinteiset “file”-toiminnot. Missä tahansa nykypäivän ohjelmassa on yleensä aina File-valikko, eli tiedoston toimintaan vaikuttava valikko (Kuva 17).



KUVA 17. Tiedoston toimintojen työskentelypalkki

Tiedoston (tai tarkemmin ohjelman) toimintaan voi Skeletor 2016 -ohjelmassa vaikuttaa viidellä eri tavalla. Ensimmäiset neljä nappia avaavat ohjelmassa ponnahtusikkunan,



jossa tarjotaan napin kuvastama toiminto. Viimeinen nappi on tarkoitettu ohjelman sulkemiseen.

### **Uusi projekti**

Ensimmäisessä napissa lukee “New”, ja se tarkoittaa lyhyesti uutta projektia. “New”-nappia painettaessa avautuu varmistusikkuna, jossa käyttäjältä kysytään, haluaako käyttäjä varmasti aloittaa uuden projektin. Jos käyttäjä painaa kyllä-nappia, koko luurangon käännetyt luut nollautuvat ja näin saadaan aloitettua täysin uusi projekti. Jos käyttäjä on esimerkiksi painanut nappia vahingossa, on mahdollisuus vastata “ei”, mikä sulkee ponnahdusikkunan tekemättä auki olevalle luurangolle muutoksia.

### **Vanhan projektin avaaminen**

Toinen nappi (“Load”) on aikaisemmin luodun ja tallennetun luurangon tarkastelua varten. Nappia painettaessa ruudulle ilmestyy ponnahdusikkuna, jossa käyttäjä voi klikata pudotusvalikkoa, johon on listattu aiemmin tallennettujen tiedostojen nimet. Käyttäjä voi valita haluamansa tiedoston ja tämän jälkeen hyväksyä valinnan napilla, jossa on “>”-symboli. Kun käyttäjä on hyväksynyt valinnan, ohjelma lukee tiedoston ja asettaa luurangon samaan asentoon, kun se on aikaisemmin tallennettu. Tämä ominaisuus ei tässä Skeletor 2016 -versiossa kuitenkaan palauta automaattisesti luiden kiertymistä indikoivaa asteittasta punaisen värin lisäämistä, vaan kaikki luut avautuvat oletusarvoisen valkoisina.

### **Projektin tallentaminen**

Kolmannessa napissa lukee “Save”, ja se tarkoittaa luurankoon tehtyjen muutoksien tallentamista. Tallenna-nappia painettaessa ruudulle ilmestyy ponnahdusikkuna, jossa käyttäjää pyydetään syöttämään tekstikenttään nimi, jolla halutaan kyseinen luuranko tallentaa. Tiedosto tallentuu .csv-tiedostomuotoon, joka pystytään myös avaamaan esimerkiksi Microsoftin Excel-ohjelmalla. Kun käyttäjä on syöttänyt nimen ja hyväksynyt,

että haluaa nimeä käyttää, ohjelma kirjoittaa taulukon, johon on listattu kaikki luut ja niiden muutetut luun asteet X-, Y- ja Z-akseleilla. Tallennetut .csv-tiedostot löytävät Skeletor 2016 -kansioista.

### **”?”-symboli**

“?”-nappi tarkoittaa apuvalikkoa, josta käyttäjä voi tarkistaa ohjelman version ja käyttövinkit. Nappia painettaessa avautuu ponnahdusikkuna, jossa on ohjeistustekstiä ohjelman käyttöä varten ja x-symbolin sisältävä nappi sulkemista varten. Ponnahdusikkunaan on myös sisällytetty lopputekstit (credits), jossa lukee ohjelman tekemiseen osallistuneiden henkilöiden nimet ja roolit ohjelmaa tehtäessä.

### **Ohjelman sulkeminen**

Viimeinen nappi, “Exit”, sulkee ohjelman. Ohjelma toimii pääasiassa koko näytön tilassa (fullscreen), jolloin ikkunan yläreunasta ei voi klikata ruksi-symbolin sisältävää nappulaa. Tällöin edellä mainittu Exit-nappi ajaa saman asian. Sulkemisnappia painettaessa ohjelma sulkeutuu välittömästi. Nappi on pyritty sijoittamaan siten, että käyttäjä ei vahingossa painaisi sitä.

## 6 POHDINTA

Olen erittäin tyytyväinen saamaani opinnäytetyöaiheeseen ja lopputuloksiin. Keskimäärin ylitin odotukseni opinnäytetyöni käytännönsuuden lopputuloksen suhteen. Opettelin kevään 2016 aikana minulle täysin uuden ohjelmointikielen, C#:n. Aiempaa ohjelmointikokemusta minulla oli ainoastaan Javasta, joten C# tuntui paljon helpommalta käyttää Javaan verrattuna. Kehitin myös syvempää tietämystä Cinema 4D:stä ja hain uutta sekä nopeampaa työnkulkutekniikkaa (workflow). Työnkulku ei ollut aluksi nopein mahdollinen, sillä kaikki mahdolliset työkalut 3D-mallinnusohjelmassa eivät vielä olleet niin tuttuja. Työn edetessä 3D-mallinnusohjelman työkalut ja ohjelman erilaiset käyttöominaisuudet alkoivat tulla enemmän tutuiksi, joten työn eteneminen nopeutui loppua kohden. Edelleenkin ei ole yhtä oikeaa tapaa toteuttaa 3D-mallin tekemistä, pitää vain löytää itselle sopivin työjärjestys. Esimerkkinä mainittakoon ylimääräisten ja irrallisten verteksien poisto. Aluksi valitsin 3D-mallin verteksit, jotka ovat yhteydessä toisiinsa, jonka jälkeen tein käänteisen valinnan ja sen jälkeen käytin poisto nappia. Ohjelmassa kuitenkin olisi ollut valmiiksi ominaisuus, joka tarkistaa onko mallissa verteksejä, joita ei ole yhdistetty muihin vertekseihin yhdellä klikkauksella.

Kun määritellään työnosuuksien suuruus prosenteissa käytetystä kokonaisajasta, niin seuraava luettelo kuvastaa hyvin käytetyn ajan jakautumista:

- 38 % 3D-mallintaminen, nimet, hierarkia
- 37 % C# metodien etsiminen ja Unityn käyttämisen selvittäminen
- 21 % ohjelmointi
- 4 % muu; 2D-grafiikka, käyttöliittymä ja sen suunnittelu, ohjelman suunnittelu

Sen lisäksi, että opin todella paljon uutta ja sain syvennettyä tietoja entuudestaan tutuissa asioissa, sain tehtyä omasta mielestäni käyttökelpoisen tuotteen. Ideana ja toteutuksena olen ohjelmaan todella tyytyväinen. Haluaisin tietysti, että tulevaisuudessa ohjelman saisi työstettyä täysin virheettömäksi ja käyttöliittymän täydelliseksi. Tällä hetkellä tiedossa on paljon parannusideoita, mutta resurssit (pääasiassa aika), ovat lopussa. Rahaa tuotteella ei voi vielä hankkia, enkä raaskisikaan ottaa rahaa vastaan, jos sitä tarjotaisiin. Skeleton 2016 versio 1.0 ei kuitenkaan vielä ole loppuun asti hiottu ohjelma. Lisäksi se on toteutettu vain yhden asiakkaan toiveiden mukaan.

Skeletor 2016 -ohjelmassa voisi parannella muun muassa yleisesti ohjelman koodin optimointia ja luurangon 3D-mallia. Käyttöliittymääkin voisi hioa vielä hienommaksi, sillä tällä hetkellä käytössä on raakakoodatut nappulat (eli koodin puolella on määritely, mitä nappulasta tapahtuu, minkä värinen ja kokoinen se on ja niin edelleen), joiden tilalle Unitylla on tarjota valmiita asetteja (assets), joilla voisi toteuttaa helposti ja nopeasti paremman käyttöliittymän. Valmiissa asetti-nappuloissa on myös se etu, että niitä voi säätää visuaalisesti juuri sellaisiksi kuin haluaa. En käyttänyt edellä mainittuja valmiita nappuloita, sillä en aluksi tiennyt niiden olemassa olosta ja olin ehtinyt työstää usean nappulan jo puhtaasti koodaamalla. Skeletor 2016 -ohjelmasta voisi tehdä myös kätevän mobiiliversion tableteille artistien käyttöön, sillä markkinoille tulee koko ajan parempia piirtotabletteja. Artistien mobiiliversiona voisi olla kattava valikoima valmiita luurangon poseerauksia, joita käyttäjä saisi itse hienosäätää.

Markkinoille suoranaisesti ”Skeletor 2016” nimi ei todennäköisesti kelpaa, sillä Skeletor-niminen hahmo on tullut tunnetuksi tutusta Mattelin lelusarjasta: ”Master of the Universe”, myöhemmin TV-sarjoista sekä elokuvasta. Skeletor-hahmo on sarjan päähahmon, He-Manin, arkkivihollinen. (Wiki Grayskull 2016.) Joten projektin jatkotyöstöä varten, myyvän nimen keksiminen ohjelmalle on prioriteettilistan kärjessä.

Minulla oli myös projektin aikana hyvin erilaisilla taustoilla olevia käyttöttestaajia, joten ohjelman potentiaali paljastui suhteellisen huikeaksi. Esimerkiksi eräs juuri sosionomiksi valmistunut totesi, että olisipa ohjelma ollut käytössä opiskelun aikana, sillä ohjelmalla luiden nimien opiskelu olisi ollut todella paljon helpompaa. Myös grafiikka-artistit totesivat, että saavat ohjelman avulla hyvän referenssin ihmistä piirrettäessä. Ohjelma on myös kustannustehokas, kun esimerkiksi fysioterapeutti haluaa havainnollistaa asiakkaalle muun muassa tämän kehon virheasentoja, eikä asiakasta välttämättä tarvitse lähettää magneettikuviin.

Näen Skeletor 2016 -ohjelman tulevaisuuden valoisana ja toivon sen myös inspiroivan muita kehittämään vastaavia ohjelmia tai sen toimivan hyvänä pohjana ihmisten terveyttä ja hoitamista edistäviin ohjelmiin.

## LÄHTEET

Franssila, P. & Wallin, M. 2010. Fysioterapia on potilaan ohjausta. Fysioterapia-lehti 5/2010, 5.

Codementor. ”What is the best programming language for me?” 2016. Codementor. Luettu 25.9.2016. <http://www.bestprogramminglanguagefor.me/why-learn-c-sharp>

Dickinson, C. 2015. Unity 5 Game Optimization. Packt Publishing. Hocking, J. 2015.

The Guerrilla CG Project. Euler (gimbal lock) Explained. Youtube 2009. Katsottu 22.8.2016. <https://www.youtube.com/watch?v=zc8b2Jo7mno>

Unity Blog. 2015. Unity Technologies. Luettu 13.9.2016. <https://blogs.unity3d.com/2015/05/28/web-publishing-following-chrome-mpapi-deprecation/>

Unity Documentation. 2016. Unity Technologies. Luettu 25.9.2016. <https://docs.unity3d.com/Manual/>

Unity Multiplatform. 2016. Unity Technologies. Luettu 25.9.2016. <https://unity3d.com/unity/multiplatform>

Unity Store. 2016. Unity Technologies. Luettu 25.9.2016. <https://store.unity.com/>

Weisstein, E. "Euler Angles." From MathWorld--A Wolfram Web Resource. Luettu 22.8.2016. <http://mathworld.wolfram.com/EulerAngles.html>

Wiki Grayskull. 2016. ”Skeletor” Fandom TV Community powered by Wikia. Luettu 10.10.2016. <http://he-man.wikia.com/wiki/Skeletor>