

Tia Falck

2D-deformaatio-animaatio peligrafiikassa

Metropolia Ammattikorkeakoulu

Medianomi

Viestintä

Opinnäytetyö

26.5.2017

Tekijä(t) Otsikko	Tia Falck 2D-deformaatio-animaatio peligrafiikassa
Sivumäärä Aika	32 sivua + 1 liite 26.5.2017
Tutkinto	Medianomi
Koulutusohjelma	Viestintä
Suuntautumisvaihtoehto	3D-animointi ja -visualisointi
Ohjaaja(t)	Lehtori Jaro Lehtonen
<p>Opinnäytetyössä tavoitteena oli esitellä deformaatio-animaation hyötyjä peligrafiikassa.</p> <p>Esimerkillisenä pelinä käytettiin pääasiassa Vanillawaren Dragon's Crownian, koska siinä yhdistyvät perinteinen sprite sheet -animaatiota käyttävä peligrafiikka ja animaatiotyö, jonka pystyisi tekemään helpommin kokonaan 2D-mesh-deformaatiota ja luurankoanimaatiota käyttäen.</p> <p>Projektityön osuudessa käytiin läpi animoidun 2D-hahmon työvaiheet kahdessa eri ohjelmassa, joissa molemmissa pystyi tekemään mesh-deformaatiota käyttäen animaatiota. Ohjelmiksi valittiin Spine Pro ja Cubism 2.1, koska ne soveltuivat peligrafiikan tekemiseen.</p> <p>Lopputuloksena oli kaksi versiota hahmosta, joilla oli erilainen riggauksen ja animaation työprosessi. Kummallakin versiolla pystyttiin tekemään samantyylistä animaatiota.</p> <p>Opinnäytetyöstä on hyötyä 3D- ja 2D-animaatiosta ja pelialasta kiinnostuneille artisteille ja animaattoreille, jotka haluavat olla tietoisia nykypäivän 2D-animaation mahdollisuuksista.</p>	
Avainsanat	2D, peli, grafiikka, animaatio, deformaatio

Author(s) Title	Tia Falck 2D Deformation Animation in Game Graphics
Number of Pages Date	34 pages + 1 appendice 26 May 2017
Degree	Bachelor of Culture and Arts
Degree Programme	Media
Specialisation option	3D Animation and Visualization
Instructor(s)	Lecturer Jaro Lehtonen
<p>The aim of this thesis was to present the strengths and benefits of using 2D deformation animation in game graphics.</p> <p>Vanillaware's Dragon's Crown game was used as an example since it uses traditional sprite sheet animation, despite that the animation style could have been achieved a lot easier if they used mesh deformation or bone based skeleton animation instead.</p> <p>In the project, we went through the process of creating an animated 2D character in two different programs; Spine Pro and Cubism 2.1. Both of these programs were chosen due to their capability to use mesh deformation animation in the creation of game graphics. The end result was two versions of a character. Although each version had a different rigging and animation process, both could be used to create similar animations.</p> <p>This thesis is useful to those artists and animators who are interested in 3D and 2D animation techniques in the game industry, and who want to be at the forefront of the possibilities with 2D animation.</p>	
Keywords	2D, game, graphics, animation, deformation

Sisällys

1. Johdanto	1
2. Mesh-deformaation esittely	2
2.1 Polygon Mesh	2
2.2 Tekstuurin deformaatio	3
3. Mesh-deformaation vahvuudet peligrafiikassa	4
3.1 Animaatiotapojen esittely Dragon's Crownin pelihahmojen avulla	4
3.2 2D-hahmojen käyttö 3D-grafiikan sijaan	11
4. Projektityön pipeline	13
4.1 Suunnittelu	13
4.2 Toteutus	14
4.3 Työkalut	16
4.3.1 Spine	17
4.3.2 Live2D: Cubism	18
4.3.3 Nima	19
4.4 Riggaus ja animointi	21
4.4.1 Spine Pro	21
4.4.2 Cubism 2.1	26
5. Pohdinta	31
Lähteet	34
Liitteet	
Liite 1. Linkki Spinen animaatioon	

1. Johdanto

Opinnäytetyö esittelee 2D-animaatiotekniikkaa, jonka suosio on pelialalla kasvussa. Työ keskittyy erityisesti mesh-deformaatio-animaatioon, jolla staattisia 2D-kuvia manipuloidulla luodaan illuusio 3D-muodosta. Opinnäytetyö on kirjoitettu pelialalla työskentelevän 2D-artistin ja -animaattorin näkökulmasta ja se on suunnattu 3D- ja 2D-animaatiosta ja pelialasta kiinnostuneille artisteille, jotka haluavat olla tietoisia nykypäivän 2D-deformaatio-animaation mahdollisuuksista.

Ensimmäisessä luvussa esitellään polygon mesh, joka on keskeinen osa 2D-deformaatio-animaatiota. Se on perustietoa 3D-opiskelijoille, mutta termistö voi olla vierasta 2D-artisteille. Photoshop on kaikkein suosituin kuvanmuokkausohjelma, joten käytän sen deformaatio-työkaluja aiheen havainnollistamiseen.

Vanillawaren Dragon's Crown -pelin hahmoanimaatiot saivat minut kiinnostumaan 2.5D-animaatiosta ja sen seurauksena 2D-deformaatio-animaatiosta. 3D-opiskelijana ja 2D-animaation ystävänä nämä hahmot pistivät heti silmäni ja halusin oppia tekemään samanlaista grafiikkaa. Dragon's Crown julkaistiin vuonna 2013, jolloin nykyään suosittu animaatio-ohjelmat kuten Spine ja Cubism olivat vasta kehitteillä. Dragon's Crownista löytyy uutta ja vanhaa animaatiotyylä ja siksi koen sen hahmojen olevan sopivia esimerkkejä siitä, miten mesh-deformaatiolla (*mesh deformation*) ja luurankoanimaatiolla saa nykyään tehtyä samantyylistä animaatiota helpommin. Toisessa luvussa esittelen miksi 2D:tä voidaan suosia 3D:n sijaan, mutta pidän sen lyhyenä ja ytimekkäänä. 3D:n rinnastaminen 2D-animaatioon jakaisi kirjoitelman helposti kahteen osaan.

Opinnäytetyön aihepiiri on tietoisesti rajattu mahdollisimman suppeaksi, eikä siinä käsitellä kaikkia mesh-deformaation tekniikoita tai pelejä, jossa niitä käytetään. Esimerkiksi League of Legends -pelin pelaajan sisäänkirjautumisen ohella esitetyt "animoidut maalaukset" ovat suurilta osin tehty mesh-deformaatiolla, käyttäen After Effects -ohjelmaa ja 3D-grafiikkaa. Näiden maalausten analysoinnissa ja tekniikoiden esittelyssä olisi tarpeeksi materiaalia toisen opinnäytetyön tekemiseen. Koin näiden tekniikoiden yksittäisen karsimisen ja esittelyn liian laajaksi kokonaisuudeksi tähän opinnäytetyöhön.

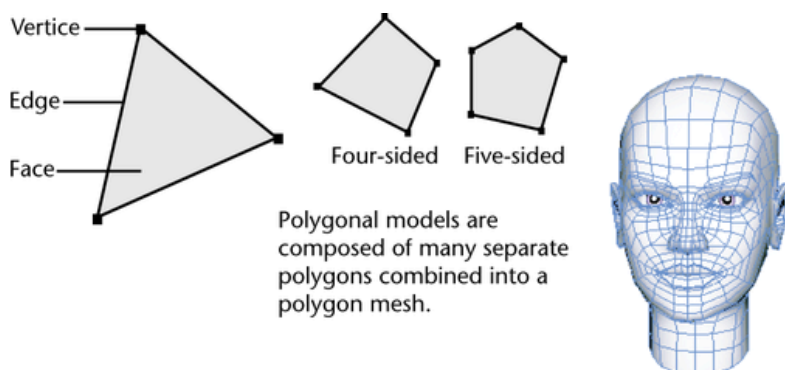
Projektityön osuudessa käydään läpi animoidun 2D-hahmon työvaiheet. Hahmossa käytetään luuanimaatiota ja mesh-deformaatiota ja se tehdään kahdessa eri ohjelmassa. Ohjelmat valittiin niiden ominaisuuksien perusteella, pääpainotteena niiden soveltuvuus peligrafiikan tekemiseen. Tavoitteena on saada aikaan hahmo, jonka rigissä käytetään mesh-deformaatiota halutun animaation tekemiseen.

Lopuksi pohdin projektityön tuloksia. Mietin mitä olisin voinut tehdä toisin ja kuinka projektista tuli ilmi 2D-deformaation hyödyllisyys peligrafiikassa, varsinkin pelihamojen animaatioissa ja mitä opinnäytetyöstä jäi käteen tulevaisuuden projektien kannalta.

2. Mesh-deformaation esittely

2.1 Polygon Mesh

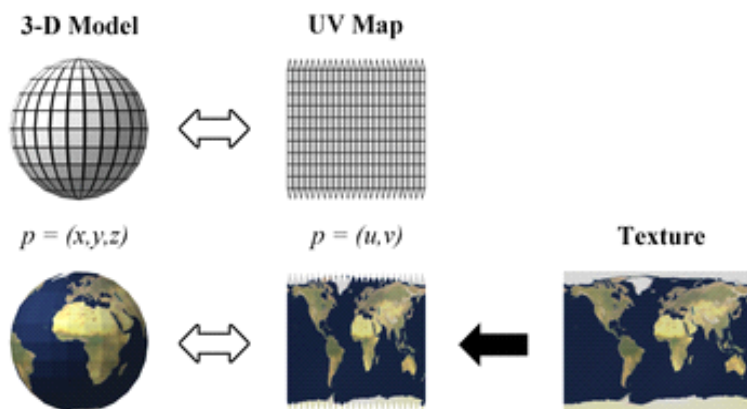
Polygon mesh (polygon verkot) on monen polygonin muodostama pinta, jolla voi visualisoida digitaalista grafiikka, kuten 3D-malleja. Polygoni on monikulmio. Vähintään kolmen pisteen eli verteksin (*vertice*) liittyessä yhteen viivoilla, joita kutsutaan edgeiksi (*edge*), muodostuu niiden sisälle planaarin face-taso. Kolmen verteksin tasoa kutsutaan kolmioksi (*triangle*) ja neljän verteksin neliöksi (*quad*). Verteksejä voi olla useampiakin eikä muoto rajoitu neliön tai kolmion rajoihin. 2D-grafiikassa verteksit pysyvät koordinaatistolla x ja y, mutta 3D-grafiikassa ne voivat sijoittua x, y ja z koordinaateille. (Autodesk Maya 2015a.)



Kuvio 1. Havainnollistavia kuvia polygonin rakenteesta. Perinteisesti polygon mesheillä on luotu 3D-grafiikka, kuten kuvassa esiintyvät kasvat. (Autodesk Maya 2015a.)

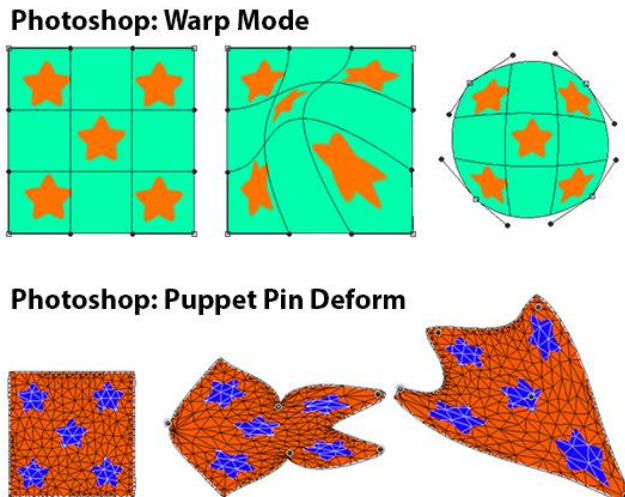
2.2 Tekstuurin deformaatio

UV-mappingillä (*UV-mapping*) tarkoitetaan 2D-grafiikan yhdistämistä 3D-meshiin. Tätä kutsutaan myös teksturoinniksi (*texturing*). Tyypillisesti UV-mappaamista käytetään 3D-grafiikan tekemisessä siten, että 3D-malli tehdään ensin, sitten siitä levitetään planaari pinta eli UV-map, johon 2D kuva yhdistetään. (Autodesk Maya 2015b.) 3D-malli teksturoidaan 2D-muodossa UV-mapilla, koska näin vältetään ei-toivottua kuvaa väärentävää deformaatiota (Wikipedia 2017).



Kuvio 2. Kuvassa 3D-mallista on tehty planaari pinta eli UV-map. (Wikipedia 2017.)

Spinessä ja muissa 2D-animaatio-ohjelmissa 2D-grafiikan deformaation työprosessi aloitetaan 3D:n työprosessiin nähden päinvastaisesti: ensin ohjelmaan tuodaan 2D-grafiikka, joko kuva atlaksen, yksittäisten kuvien tai Photoshop-tiedoston kautta, jonka päälle luodaan polygon mesh.



Kuvio 3. Photoshopin Warp Mode -työkalu luo neliötasoista koostuvan meshin 2D-kuvalle, jota voi deformaoida verteksin ja edgien positiota liikuttaen. Alempana on Photoshopin puppet pin deform -työkalulla tehtyä deformaatioita, jossa on ylempää tiheämpi verkko.

Meshin deformaatiota esiintyy 3D-ohjelmien lisäksi myös kuvankäsittelyohjelmissä. Adobe Photoshopin Liquify filter, Warp Mode ja Puppet Pin Deform ovat paljon käytettyjä työkaluja niin kuvanmuokkauksessa kuin peligrafiikan maalauksessa. Kuvio 3. selventää mesh-deformaatiota erityisesti niille, jotka Photoshop-ohjelmaa ovat aikaisemmin käyttäneet. Photoshopin puppet pin deform -työkalu luo kolmioista koostuvan meshin, joka antaa enemmän kontrollia tekijälle, koska sen avulla meshiin voi laittaa omia "pin" kontrollipointteja. Meshin kolmioverkon tiheyden lisääminen tekee sen deformatiosta luonnollisemman näköistä ja kun polygon-verkko on tarpeeksi tiheää, pinnan tekstuurin tähdet eivät menetä niille ominaista muotoaan.

3. Mesh-deformaation vahvuudet peligrafiikassa

3.1 Animaatiotapojen esittely Dragon's Crownin pelihahmojen avulla

Dragon's Crown on Vanillaware -pelistudion vuonna 2013 julkaistu fantasia teemainen 2D-side scroller -peli. Vanillaware on tunnettu käsinmaalatunnäköisistä ja erittäin yksityiskohtaisista 2D-grafiikoistaan. Käsittelen seuraavia digitaaliseen 2D-animaatioon liittyviä animaatiotekniikoita Dragon's Crownin hahmojen animaatioita analysoiden. Vanil-

lawaren hahmojen tekemiseen käytetystä teknologiasta ei ole julkaistu paljon tietoa ainakaan englanniksi, joten joudun käyttämään analyysin pohjana paljon omaa kokemustani 2D-animaatiosta. Koska hahmot on maalattu niin yksityiskohtaisesti ja öljyvärimäisellä tyyllillä, niiden animaatio on haastavaa varsinkin perinteisillä menetelmillä. Yritän antaa syytä sille, miksi luurankopohjainen animaatio ja mesh-deformaatio olisivat paras tapa Dragon's Crownille tyypillisen animaation tekemiseen.



Kuvio 4. Kuvakaappaus Dragon's Crow -pelistä. Kuvassa näkyy pelille ominainen taidetyyli ja neljä pelattavaa hahmoa: Sorceress, Dwarf, Fighter ja Amazon. (Wikipedia n.b.a)

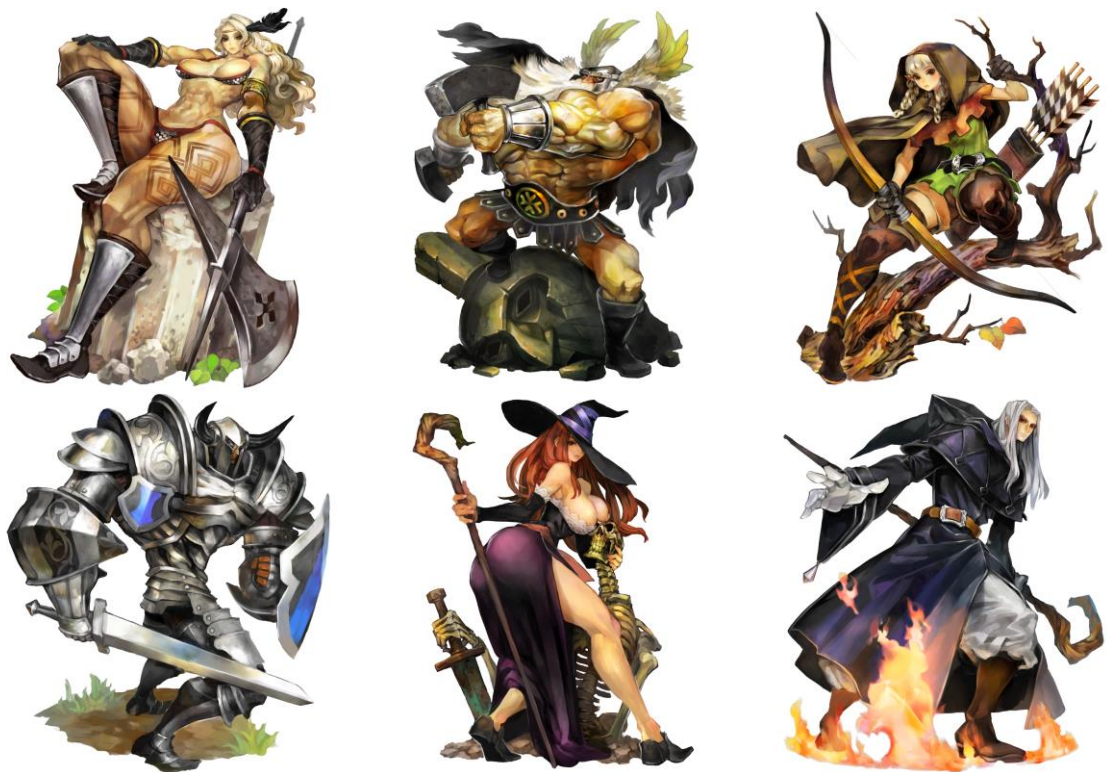
Löysin yhden haastattelun, jossa Vanillawaren pääartisti George Kamitami kertoo heidän käyttävän omatekoista Adobe Flash-tyylistä animaatio-ohjelmaa. Kamitami tekee hahmojen animaatioiden tärkeimmät keyframeit (*keyframe*) ja antaa sitten muiden artistien tehdä lopun. He käyttävät "tebineri" eli hand-shapingiksi (*hand shaping*) kutsuttua tekniikkaa, jolla hahmot saadaan näyttämään 3D:ltä. (Sheffield n.d.) Hahmojen tekstuuria "käsityönä" deformaation voidaan luoda illuusio sen 3D:mäisyydestä. Tätä tyyliä kutsutaan myös 2.5D-animaatioksi. 2.5D-animaatiolla voidaan tarkoittaa 2D-grafiikkaa, jota siirrellään 3D-tilassa tai kuten tässä tapauksessa, animaatiota jossa on käytetty erilaisia keinoja, jotta 2D-kuva saataisiin näyttämään 3D:ltä (Adrien-Luc n.d.).

Frame-by-frame on animaatiotapa, jossa jokainen frame eli kuva on piirretty erikseen. Vanhat käsin piirretyt animaatioelokuvat on tehty tällä tekniikalla, jossa yhden sekunnin kestävä animaatio vaati 24 eri kuvaa. (Bloop Animation n.d.)

Keyframe-animaatiossa hahmon animaation liikkeen vaikuttavimmat kohdat tehdään ensin. Esimerkiksi kaksi keyframeä olisi kaksi kuvaa: yksi, jossa hahmo istuu ja toinen, jossa

hahmo on noussut ylös seisomaan. Näitä kahta eri asentoa kutsutaan myös keyposeiksi (*keypose*)(Aim n.d.a). Traditionaalisessa frame-by-frame animaatioissa artisti joutuu piirtämään jokaisen näiden kahden keyframin välisen kuvan. Tämä vaatii paljon työtä ja aikaa, sekä tietynlaista taiteellista osaamista, mikä tekee frame-by-frame tekniikasta hidasta ja haasteellista.

Animaattorin näkökulmasta Dragon Crownin hahmot ovat haaste, jossa ongelmana on niiden maalauksellinen taidetyyli ja sen yhtenäisenä pitäminen kaikissa hahmojen animaatioissa. Pelissä on kuusi erilaista pelattavaa päähahmoa ja paljon erilaisia vihollishahmoja. Jokaisella hahmolla on erilainen ase ja useampi hyökkäystyyli ja niiden animaatioiden kuvakulmat poikkeavat myös toisistaan. Enimmäkseen takaapäin kuvattu lihaksikas Amazon ja panssaroitu Fighter ovat kummatkin lähitaistelijoita, mutta eroavat toisistaan niin grafiikan kuin animaatioiden perusteella selvästi: Fighterin animaatiot ovat Amazoniin verrattuna hitaita ja painavia ja hän käyttää miekan lisäksi kilpeä kun taas Amazonin hyökkäykset ovat dynaamisesti poseerattuja nopeita potkuja ja lyöntejä ja hänen suuri pertuska näyttää kevyeltä hänen käsissään.



Kuvio 5. Dragon's Crownin kuusi kontrolloitavaa hahmoa. Jokaisella hahmolla on paljon animoitavia osia ja liikkuvaa pintaa kuten kangasta ja hiuksia. (The Spriters Resource n.d.)

Cutout animation eli pala-animaatiolla hahmon jokainen liikkeen vaativa osa, esimerkiksi reisi, pohje ja jalkaterä, paloitellaan erillisiksi osiksi. Tämä tarkoittaisi sitä, että hahmon jalkaa pystyisi animoimaan kolmessa eri osassa eikä niiden liike vaikuttaisi muuhun hahmon grafiikkaan. Esimerkiksi Sorceressin sprite sheetistä (kuvio 7.) voi nähdä kuinka hiukset on paloitettu useampaan osaan yhtenäisen kuvan sijasta, jotta osia on pystytty animoimaan erikseen ja niiden liikkeeseen on saatu syvyyttä. Dragon's Crownin hahmoissa on monta liikkuvaa osaa pelkkien raajojen ja aseiden lisäksi. Pitkät hiukset, hulmuava kangas, kiiltävä metalli ja teksturoidut pinnat kuten Amazonin jalkojen tatuoinnit voivat olla haasteellisia animoida ilman paloittelua. Ilman pala-animaatiota ja mesh-deformaatiota, artistit joutuisivat maalaamaan animaation jokaisen kuvan ainakin osittain uudelleen. Dragon's Crownin maalauksellinen taidetyyli olisi käsin animoituina vaikea pitää yhtenäisenä jokaisessa kuvassa ja pienetkin virheelliset eroavaisuudet hahmon tekstuureissa pistäisivät silmään animaatioissa. Liikkuvassa kankaassa ja hiuksissa tämä voisi olla vielä uskottavan näköistä, mutta vastaavanlainen tahattomasti luotu epätasaisuus hahmon ihossa olisi häiritsevää.

Sorceress hahmon sprite sheettiä (Kuvio 7.) tutkiessani kiinnitinkin siis erityisesti huomiota hahmon hiuksiin ja polvien taipumiseen. Kummatkin näyttävät olevan staattisia kuvia, joiden grafiikka ei muutu taivutuksen lisäksi huomattavasti, toisin kuin hahmon hameessa. Saman jalan animointi käsin ei tunnu mielekkäältä, varsinkin kun sen liikuttaminen mesh-deformaation avulla on paljon helpompaa ja nopeampaa. Tämä vahvistaisi käsitystäni siitä, että aikaisemmin mainittu "tebineri" tarkoittaa Vanillawaren kehittämän ohjelman versiota mesh-deformaatiosta.



Kuvio 6. Dragon's Crownin viholliset ovat myös yksityiskohtaisesti maalattuja ja monista liikkuvista osista koostuvia, kuten kuvan Red Dragon. Mielestäni tässä kuvassa tulee hyvin esille kuinka mahdottomalta projektilta tämän kaltaisen hahmon animoiminen olisi pelkästään frame-by-frame tekniikalla. (Vanillaware n.d.)

Sprite sheet (*sprite sheet*) toimii frame-by-frame -animaation tapaan, eli jokainen muutos hahmon liikkeessä vaatii oman kuvansa. Yksittäiset kuvat animaation liikkeistä ovat yhdellä tekstuurilla, josta niitä käyttävä peli näyttää yhden kuvan kerrallaan. Kun peli näyttää peräkkäiset kuvat yksitellen tarpeeksi nopeasti, luodaan illuusio liikkeestä (Lambert 2016).

Dragon's Crownin pelimoottori käyttää sprite sheettejä (*sprite sheet*) animaatioiden pyörittämiseen. Tämän pystyy näkemään hahmojen animaatiota tarkastellen sekä Dragon's Crownin Wikia -sivuston sprite sheeteistä. Sprite sheet on perinteinen tapa käsitellä hahmojen animaatiota peleissä. Hahmot voi luoda ja animoida eri ohjelmissa, jopa 3D-malleina, ja niiden animaatiot exportataan (*export*) PNG.-kuvasarjoiksi, jotka asetellaan sprite sheetille. Esimerkiksi Ori and the Blind Forest -pelissä päähahmo Ori on 3DsMaxilla luotu ja animoitu 3D-malli, jonka animaatiot on rendattu (*rendering*) ohjelmasta 2D-kuviksi ja sprite sheeteiksi (Benson 2015).

Sprite sheetin merkittävin ongelma on sen vaatima muistitila: korkealaatuinen grafiikka vaatii suuret mittasuhteet ja jos yhdestä hahmosta on vähintään 12 eri kuvaa per animaatio, voi jo pelkästään hahmojen vaatima muistitila olla liian suuri joihinkin projekteihin. (Codrina 2016.) Kuvien määrä per animaatio vaikuttaa animaation laatuun ja mitä

enemmän niitä on, sitä sulavamman ja yksityiskohtaisemman hahmon animaatiosta saa. Jos hahmojen sprite sheeteistä pitää vähentää kuvia muistitilan säästämiseksi, animaation sulavuus kärsii nopeasti. Tämän ongelman huomaa varsinkin silloin jos pelissä on kohtauksia, jossa kaikki pelin animaatiot hidastuvat: Dragon's Crownista otettuja videoita manuaalisesti hidastamalla huomaa, kuinka hahmojen liikkeet muuttuvat katkonaisiksi kun sprite sheetin kuva vaihtuu liian hitaasti toiseksi.



Kuvio 7. Sorceress hahmon 12 kuvan kävelyanimaation sprite sheet. Kuva animoidaan vasemmalta oikealle ja jatkuu samoin seuraavalla rivillä. Viimeisen kuvan jälkeen animaatio alkaa alusta. (Dragon's Crown Wikia, n.d.a)

Nykyaikainen vaihtoehto sprite sheeteille on luuanimaatio. 2D-animaatiossa käytetty luuranko on muuten samanlainen kuin sen 3D-versio, mutta sen osat voivat liikkua vain x- ja y-akseleilla. Luurankoanimaatiossa täytyy käyttää pala-animaatiota, eli hahmo pitää leikata osiin. Hahmon osat kiinnitetään luurankoon kuvina ja niitä animoidaan luiden avulla. Hahmon animaatio koostuu luiden positioiden keyframeistä. Luuranko ja sen animaatiot voi suoraan siirtää pelimoottoriin ilman sprite sheetiä, mikä säästää tilaa. Luurankoanimaatio on erittäin yleinen animaatio-tekniikka peligrafiikassa. (Dooby 2014.)

Keyframejä käyttäen hahmon saa nopeasti liikkeelle: toisin kuin frame-by-frame animaatiossa, luilla animoidun hahmon animaation keyframien välit ovat tietokoneen piirtämiä. Tätä kutsutaan tweenaukseksi (*tweening tai inbetweening*). Mesh-deformaatiossa tietokone tekee myös meshin kahden keyframen välisen deformaation.



Kuvio 8. Hahmo, jonka luurangon luut on värikoodattu, jotta niihin linkitetyt palat on helpompi tunnistaa. Varsinkin jaloissa tämä helpottaa oikean luuketjun valitsemista.

Dragon's Crownin hahmojen isojen eroavaisuuksien takia yhden hahmon animaatio on vaikea suoraan kopioida toiselle. Luuanimaatiossa luuranko voi pysyä samana, vaikka siihen kiinnitetyt kuvat muuttuisivat, eli jos kaksi hahmoa pystyy jakamaan saman luurangon, niiden animaatiot voi myös jakaa keskenään. Saman luurangon käyttäminen voi säästää aikaa animaation kannalta. 2D-luurango on kuinkin varsin nopeasti luotavissa nykypäivän animaatio-ohjelmilla kuten Spinellä (katso luku 4.), varsinkin jos ei käytä mesh-deformaatiota. Koska mesh-deformaatio tapahtuu kuvassa, jonka päälle on luotu polygon mesh, sen animaatiota ei välttämättä pysty siirtämään suoraan muille hahmoille. Kuvan täytyy pysyä meshin sisällä eli jos alkuperäinen mesh on tehty esimerkiksi lyhyille hiuksille, ei sitä voi kopioida toisen hahmon pitkille hiuksille.

Yksi sprite sheetin vahvuuksista on se, että hahmon koko animaatiota pystyy muokkaamaan kerralla. Esimerkiksi metallin kiillon voi maalata Photoshopissa suoraan sprite sheetille frame-by-frame tyylisesti. Tässä auttaa Photoshopin ja muiden digitaalisten kuvanmuokkausohjelmien layer-blendaukset (layer blend mode) kuten esimerkiksi Multiply-mode, jolla varjoja voi tummentaa ja Screen-mode, jolla voi maalata hohtoa ja valoa. Tämä voi olla yllättävän nopea tapa verrattuna luuanimaatioon, sillä metallin kiillon tekemisen helppous on luuanimaatiossa hyvin paljon animaatio-ohjelmasta kiinni, koska monista voi puuttua kokonaan Photoshopin layer blend moodin kaltaiset vaihtoehdot ja kiiltojen animoimiseen pitää keksiä muut keinot. Sprite sheetin animaatioon voi

helposti myös lisätä pieniä yksityiskohtia, esimerkiksi hahmon silmien räpytyksen tai kasvojen ilmeen muutoksen, jotka kummatkin vaativat luuanimaatiolta enemmän työtä.



Kuvio 9. Fighter -hahmon kävelyanimaation keyposeja, joissa näkyy kuinka panssarin metallia on tummennettu ja vaalennettu valon ja varjon muutosten luomiseksi. Muutos on paljon helpommin havaittavissa animaatiossa kuin kuvasarjassa. (Dragon's Crown Wikia. n.d.b)

Toivottavasti Vanillaware ottaa tulevaisuudessa projekteissaan kaiken irti nykypäivän luu- ja mesh-deformaatio-animaatioon keskittyvistä ohjelmista. Sprite sheetit rajoittavat heidän kauniin taiteen sulavuutta ja uskon että he pystyisivät tuottamaan vielä laadukkaampaa animaatiota ja taidetta, kunhan työkalut ovat oikeat.

3.2 2D-hahmojen käyttö 3D-grafiikan sijaan

Harvoin 2D-hahmo muuntuu 3D-hahmoksi, ilman että alkuperäinen versio muuttuisi: kuviossa 10. näkyy selvästi kuinka pelkästään kameran liikuttaminen eri asentoon saattaa muuttaa hahmon ulkonäköä merkittävästi: hahmolle ominaisen hiustyylin siluetti ja muoto muuttuu erilaiseksi pelimoottorin kameran perspektiivin takia. Tämä on myös erityisen helposti nähtävissä sarjakuvamaisessa tyylissä, jossa kasvojen piirteet kerrotaan ääriiviivoin ja tyyliteltyllä valolla ja varjoilla. Oikealla olevan hahmon kasvat (Kuvio 10.) on Live2D:n animaatio-ohjelmien kehittäjien mielestä miellyttävämmät, koska hahmon

kasvojen animaatiot ja muoto pysyvät uskollisina alkuperäisen 2D-grafiikan tyyliille (Ikinamo 2014).



Kuvio 10. (Oikealla) Live2D:n ohjelmalla animoidut 2D-kasvot 3D-hahmon vartalolla verrattuna kokonaan 3D:nä tehtyyn hahmoon (vasemmalla). Oikealla olevan hahmon koko mallin varjot ja valaisu on 3D-maailmassa luotuja, kun taas vasemmalla nenän ja huulten alla olevat varjot on tekstuuriin maalattuja ja siten staattisia.

Jotkut pelit teettävät 3D-mallien lisäksi 2D-versiot hahmoistaan. Näistä peleistä esi-merkkinä on Wispfire-pelistudion peli Herald the Interactive Period Drama. Pelimaailma ja sen hahmot on kaikki luotu 3D:nä, mutta kun hahmot puhuvat pelaajalle, 3D-mallien sijaan käytetään Live2D:n Cubism-ohjelmalla animoituja 2D-kuvia. Syynä oli suurelta osin resurssien puute; heillä ei pienenä pelifirmana ollut varaa laadukkaaseen 3D-hahmolle tehtyyn kasvoanimaatioon. Koska Herald -pelin pääpaino on sen juonessa ja hahmoissa, he kokivat kasvoanimaation olevan erittäin tärkeä niin tarinankerronnan kuin hahmoihin samaistumisen kannalta. Live2D Cubism antoi heidän yhdistää vahvan 2D-taiteen osaamisen ja 3D-animaation elävyyden ilman perinteisen frame-by-frame animaation massiivista työmäärää. (Wispfire 2016.)

4. Projektityön pipeline

Olen enimmäkseen käyttänyt vain yhtä animaatio-ohjelmaa ja työprosessia. Halusinkin siis kokeilla muita animaatio-ohjelmia ja projektityön pohjalta miettiä, mitä eroavaisuuksia niillä on ja miten niillä saa tehtyä 2D-deformaatiota. Päätin jo ennen projektin aloittamista, että teen ihmishahmon. Tässä luvussa esittelen projektityöni vaiheet: miten suunnittelin hahmoni, toteutuksen ja miksi valitsin käyttämäni ohjelmat, miten tein tutut menetelmät toisin ja mitä ongelmia kohtasin. Projektityö keskittyy enemmän hahmon riggaamiseen ja rajaa animaatiovaiheen selostuksen hyvin suppeaksi.

4.1 Suunnittelu

Pipeline aloitetaan animoitavan hahmon suunnittelulla. Tulevia työvaiheita ajatellen on tärkeää olla jo heti alusta asti tietoinen siitä, mihin tarkoitukseen hahmoa käytetään ja kuinka paljon sitä tullaan animoimaan. Kokenut artisti voi maalata kuvan nopeasti, mutta jos hänellä ei ole mitään tietoa siihen tulevan animaation vaatimuksista, saattaa hahmon joutua maalaamaan uudestaan useamman kerran, mikä tuhlaa aikaa. Projektin laajuus ja sille suunniteltu aikataulu ovat ratkaisevia tekijöitä ja alusta lähtien.

Projektia suunnitellessa päätän onko hahmoni lähempänä kevyesti animoitua 2D-kuvaa vai pelin maailmassa uskottavasti liikkuva sekä pelaajan ohjaukseen sulavasti reagoiva pelihahmo. Animoidulla 2D-kuvalla tarkoitan hahmoa, jonka animaatio ei muutu niin paljon, että se liikkuu merkittävästi alkuperäiseltä paikaltaan. Vaikka pelihahmo ja animoitu 2D-kuva käyttäisivät samaa kuvamateriaalia, ne eroavat selvästi toisistaan animaation kannalta: pelihahmolla pitää pystyä tekemään pelin vaatimia liikkeitä, esimerkiksi juoksemaan, uimaan ja hyppimään kun taas animoidussa 2D-kuvassa hahmo pysyy enimmäkseen aloillaan eikä sen tarvitse olla pelaajan kontrolloitava. Pelihahmossa rigin (*rig*) eli luurangon täytyy olla sen verran joustava, että sillä pystyy tekemään monta hyvin toisistaan poikkeavaa animaatiota, kun taas animoidussa 2D-kuvassa kannattaa riggata (*rigging*) ja erotella palasiksi vain ne osat, joihin tulee liikettä. Riggaaminen tarkoittaa kontrollien osien luomista, joilla hahmoa voi animoida, tässä tapauksessa siis luurangon luiden ja polygon meshien luomista. Jos pelihahmo animoidaan juoksemaan ja kävelemään, erityisesti jalkojen alueen suunnitteluun tulee kiinnittää huomiota: kuinka monesta

osasta hahmon jalat koostuvat? Täytyykö jalat maalata erikseen edestä ja sivusta, jos hahmo kääntyy kameraan päin? Mikäli hahmolla on päällä mekko tai paljon erillisiä animoitavia osia, saattaa yksinkertaisenkin kävely-loopin animoiminen viedä odotettua kauemmin. Pääsääntönä kannattaakin pitää se, että mitä enemmän hahmossa on liikkuvia osia, sitä kauemmin sen animaation tuotannossa menee.



Kuvio 11. Toiseen projektiin tehdyn pelihahmon asusteen konseptit, joista valitessa mietin erityisesti hahmon jalkojen animaation ja hameen designin vaikutuksia toisiinsa.

Oman projektityöni kohdalla päätin pelihahmon tekemisen sijaan yrittää kopioida Live2D:n ohjelmilla tehtyjä animaatioita, joissa ihmishahmo seisoo paikallaan ja sen animaatio keskittyy enemmän kasvoihin, hiuksiin ja kehon pienimuotoiseen liikkeeseen. Luokittelisin siis tämän animoiduksi 2D-kuvaksi. Taidetyyli olisi Wispfire studion Herald-pelin 2D-hahmojen inspiroima, mutta hieman yksinkertaisemmin maalattu. Kasvojen piirteillä pitäisi saada tehtyä useampia ilmeitä ja hahmon nenän deformaation avulla voisin kokeilla yrittää saada kasvot kääntymään hieman sivulle ja jopa ylöspäin, jotta niille saisi 3D-mäisen vaikutelman. Tätä 2.5D-effektiä voisin kokeilla myös hahmon kehossa..

4.2 Toteutus

2D-grafiikan digitaaliseen maalaamiseen on monta ohjelmaa. Itse käytän Adobe Photoshop kuvankäsittelyohjelmaa, koska sen Photoshop-tiedostot voi yleensä exporttaa eli tuoda suoraan animaatio-ohjelmiin. Peliprojektissa saman ohjelman käyttäminen helpottaisi yhteistyötä muiden artistien kanssa, kun kaikilla on sama tiedostomuoto. Ensimmäisessä luvussa mainitsemallani Photoshopin puppet warp toolilla voi kokeilla hahmon osien deformaatiota jo maalausvaiheessa ja näin jo hieman

testata kuinka helposti hahmon osat saisi animoitua myöhemmin.

Annan itselleni viisi tuntia aikaa hahmon visuaaliseen suunnitteluun ja maalaamiseen. Aikaisempien projektien kautta olen oppinut, että varsinkin monimutkaiset hahmot, kuten ihmis- ja eläinhahmot, joissa on monta liikkuvaa osaa, on paras maalata ensiksi yhtenä kokonaisuutena. Näin valmiiksi maalatusta hahmosta voi leikata vain ne osat joita tullaan tarvitsemaan animaatioissa eikä tiedosto ole täynnä turhia layereita. Tietysti osat, jotka tietäisin jo alun perin olevan eri layer-tasoilla, kuten esimerkiksi hulmuava viitta, pitäisin jo alussa oikealla paikalla hahmon takana koska se helpottaisi maalaamista.



Kuvio 12. Hahmon kasvat ja hiukset maalattiin yhtenä kuvana ja vasta kun olin tyytyväinen lopputulokseen, leikkasin erillisiksi kuviksi tarvitsemäni palat.

Hahmon kummatkin käsivarret olivat aluksi suorat ja osittain kuvatiedoston rajojen ulkopuolella, mutta asento näytti liian jäykältä omaan makuuni ja kummankin käden animaatio olisi helposti jäänyt hyvin samanlaiseksi, joten siirsin toisen käden hahmon lanteelle. Käsien maalaus ei ole vahvin osaamiseni, joten uudelleen maalauksessa meni vähän liiankin kauan aikaa ja ylitin viiden tunnin aikarajani. Hahmojen kädet ovat kolmessa osassa: käsivarsi, kädensuojus ja käsi. Kädensuojus on peittämässä käden ja käsivarren palojen rajaa ja toivottavasti antaa näin kädelle enemmän liikkumavaraa. Normaalisti olisin pitänyt hahmon käsivarren kahdessa palassa, jotta sitä pystyisi taivuttamaan vapaasti, mutta halusin nähdä, pystynkö käyttämään palojen sijaan käden taivutukseen pelkästään mesh-deformaatiota. Pelkäsin pitkien hiusten animaation ja riggaamisen kestävän liian kauan, joten päädyin lyhyeen leikkaukseen, jossa etuhiukset ovat kolmessa

osassa ja takahiukset kahdessa. Maalasin hahmon vaatteisiin yksityiskohtia, joiden pää-tarkoitus on auttaa luomaan 2.5D-efekti kehon deformaatioissa sillä niiden liike on selvästi huomattavissa.

Jotta kasvat näyttäisivät luonnollisilta, sen palojen pitää sulautua ihoon: palojen rajat ovat osittain läpinäkyvät, nenän vartta lukuun ottamatta. Lopullinen hahmo koostuu 28 palasta, joita olisi voinut olla paljon enemmänkin, mutta päätin pitää hahmon yksinkertaisena. Jouduin myöhemmin palaamaan tiedostoon leikkaamaan hahmon kasvoista useampia osia: tarvitsin silmän luomet erillisiksi toisistaan toisen ohjelman kokeilua varten.



Kuvio 13. Valmiiksi maalattu ja paloiteltu hahmoni, joka on valmis seuraavaa vaihetta varten

4.3 Työkalut

Tässä luvussa esittelen lyhyesti kolme ohjelmaa, joista valitsin kaksi projektityöhöni. Halusin ottaa selville mitä eroavaisuuksia ohjelmien työkaluilla ja työprosesseilla on ja mikäli ne ovat nopeampia ja tehokkaampia, kuin mihin olen itse tottunut Spine Prolla työkennellessäni. Ohjelmaa valitessa on tärkeää tietää sen vahvuudet ja heikkoudet ja kuinka ne sopivat omiin tavoitteisiini.

Animaatio-ohjelmien tärkein ominaisuus on ensisijaisesti niiden soveltuvuus peligrafiikan tekemiseen. Tarkoitus on löytää ohjelma, jonka animaatioiden käyttö peleissä ei rajoitu

sprite sheeteihin. Käytännössä tämä tarkoittaa sitä, että ohjelmilla täytyy pystyä tuottamaan hahmoja, joiden luiden ja animaatioiden datan pystyy helposti exporttaamaan (*export*) eli viemään pelimoottoriin kuten esimerkiksi Unityyn, missä niitä voi nopeuttaa, hidastaa tai muuten muokata enemmän peliin sopiviksi. Tämän takia ohjelmat kuten Photoshop ja After Effects, joissa kummassakin voi tuottaa animaatiota 2D-deformaatioilla, eivät soveltuneet tähän projektityöhön. After Effectsin suurin ongelma on siinä, että sillä tehdyt animaatiot voi exporttaa vain PNG-kuvasarjoina tai videona.

Toinen syy monien ohjelmien pois karsimiseen oli niiden hinta. Olen aikaisemmin ostanut Spinen Pro version, jonka hinta oli noin 300 euroa. Tähän asti olen Spinellä pystynyt tekemään kaiken tarvitsemani, joten en nähnyt syytä investoida rahaa muiden ohjelmien kokeilemista varten. Vaikka monilla ohjelmilla on ilmainen kokeiluversio, niiden rajoitukset eivät toimineet projektini kannalta: esimerkiksi Spinen, Creaturen ja Mohon kokeiluversiossa projektia ei pysty tallentamaan, eikä niistä pysty exporttaamaan ulos mitään materiaalia. Koska opinnäytetyöni rajoittuu pelkästään animaatioon ja sitä edeltäviin työvaiheisiin, eikä hahmon ja sen animaatioiden käyttämiseen pelimoottorissa, halusin ohjelman pystyvän exporttaamaan projektista vähintään videota tai animoitua gif.-kuvaa.

4.3.1 Spine

Esoteric Softwaren kehittämä Spine on vuonna 2013 Kickstarterin avulla luotu 2D-animaatio-ohjelma. Se keskittyy 2D-luuanimaatioon ja se on ensisijaisesti kehitetty pelintekijöille. Esoteric Softwaren sivuilla on paljon tietoa ja tutoriaaleja ohjelman käyttämisestä ja foorumeilla käyttäjät voivat jakaa omia töitään sekä neuvoa toisiaan. (Esoteric Software n.d.)

Spinestä on kaksi erilaista versiota: Spine Essentials ja Spine Pro (Professional). Spine Essentialissa voi tehdä vain luupohjaista pala-animaatiota ja suurin osa muista työkaluista on vain Spine Pro versiossa, esimerkiksi IK-kahvat ja meshit.

Kokeilin ohjelmaa ensimmäisen kerran keväällä 2014 ja se on kehittynyt paljon viimeisten vuosien aikana. Käytän itse Spine-Pro ohjelmaa melkein kaiken animaation tekoon

ja monien ohjelmien kokeilemisen jälkeen koen sen nykyään yhdeksi parhaimmista peligrafiikkaan painottuvista animaatio-ohjelmista. Spinen animaatiot saa suoraan exportattua Unityyn ja muihin pelimoottoreihin.

4.3.2 Live2D: Cubism

Live2D Inc. on japanilainen yritys, joka keskittyy tuottamaan 2D-grafiikkaan perustuvia animaatio-ohjelmia. Live2D:n menestyneimmät ohjelmat; Cubism ja Cubism 2 julkaistiin vuonna 2014 ja Cubism 2.1 vuonna 2016. Uusin versio ohjelmasta, Cubism 3, julkaistiin 26. maaliskuuta 2017. Opinnäytetyöhön nähden myöhäisen ajankohdan takia, en käsittele tätä uusinta versiota.

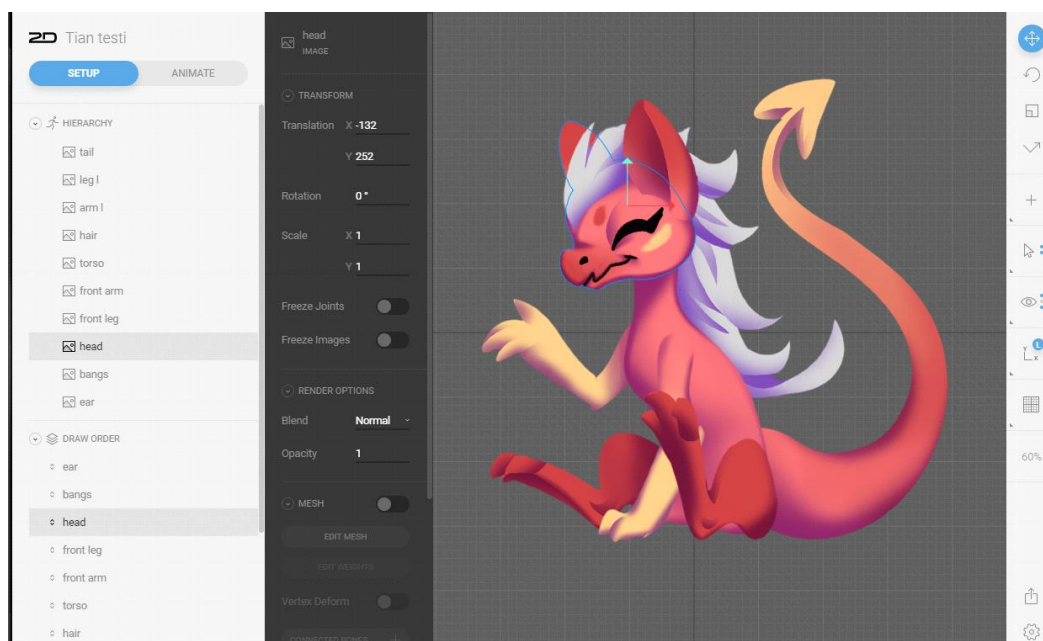
Cubism on 2D-animaatio-ohjelma, jolla on tehty satoja animaatioita peleihin, esityksiin ja Aasiassa suosittuihin visuaalisiin novelleihin (*visual novels*). Ohjelman tavoitteena ja vahvuutena on säilyttää alkuperäisen 2D-taiteen visuaalinen tyyli (Live2D Inc. n.d.a). Live2D:n sivuilla esimerkeiksi listattujen pelien perusteella ohjelmaa käytetään eniten hahmoanimaatiossa, varsinkin japanilaisen sarjakuvan tyyllisen grafiikan animoinnissa, joissa animaatio painottuu hahmojen kasvoihin ja kehon pieniin eleisiin.

Live2D: Cubism 2.1 jakautuu kahteen eriliseen ohjelmaan: Cubism Modeler, jossa hahmo rigataan ja Cubism Animator, johon valmis työ viedään animoitavaksi. Toisin kuin muissa deformaatio-animaatiota tarjoavissa ohjelmissa kuten Spinessä ja Nimassa, Cubism 2.1:ssä ei käytetä luuanimaatiota. Cubismissä käytetään ensisijaisesti mesh-deformereita ja niiden muutoksien tallentamista motion parametreille (*motion parameters*) (Live2D Inc. n.d.b). Live2D:n Cubism 2.1 ilmainen kokeiluversio oli sopiva valinta projektilleni Spinen rinnalle, koska rajoitteistaan huolimatta, sillä saa aikaan sen, mitä projektini vaatii.

Cubismin toiminnoista löytyy eniten tietoa japaniksi ja ohjelman käyttöliittymästä löytyy paikoitellen japaninkielistä tekstiä. Live2D on alkanut tekemään enemmän englanninkielistä materiaalia ohjelman toiminnoista, mutta vielä suurin osa ohjelman toimintojen dokumentoinnista on vain japaniksi. (Live2D Inc. n.d.b.)

4.3.3 Nima

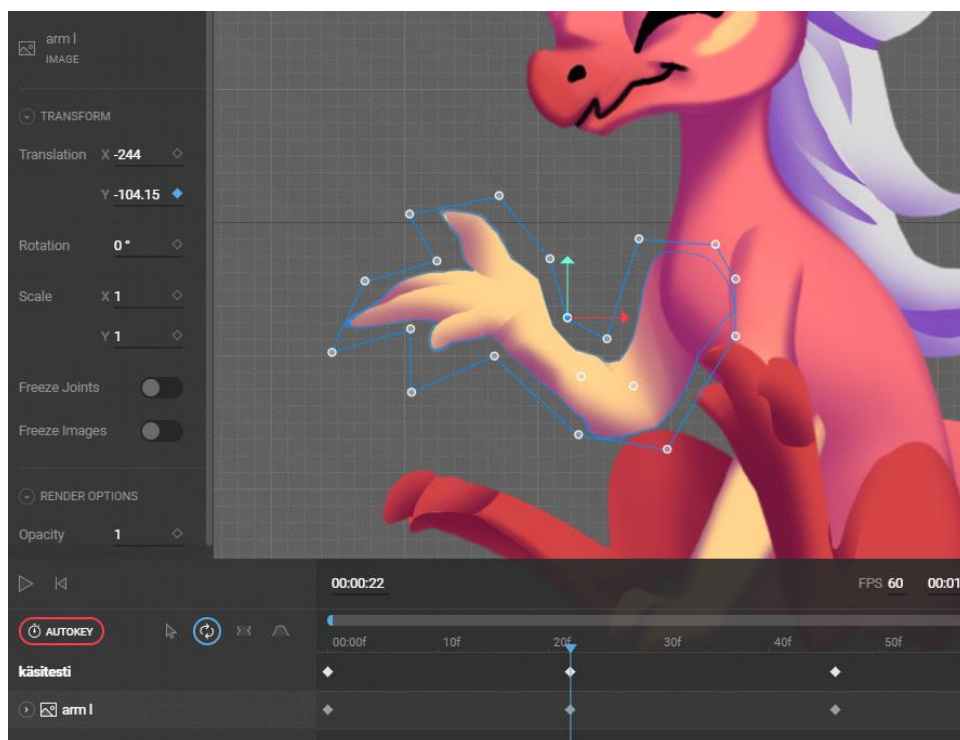
Spine ja Live2D Cubismin rinnalle löytyy kolmas ehdokas, jolla on hyvin yksilöllinen ominaisuus: 2Dimensionsin Nima on ainakin tämän opinnäytetyön kirjoitushetkellä ainoa kokonaan pilvipalvelussa toimiva 2D-animaatio-ohjelma. Se julkaistiin rajattuna beta-testauksena vuoden 2016 syksyllä. Nykyään Niman beta-versiota pystyy kokeilemaan vapaasti ja ohjelma on ainakin beta-testauksen aikana ilmainen (Rosso n.d.).



Kuvio 14. Niman käyttöjärjestelmä on käyttäjäystävällinen ja toimiva. Vasemmalla on listana projektin materiaalit, keskellä tummalla palkilla on valitun assetin tiedot ja vasemmalla on valikko, josta voi valita työkalut. Oikeasta yläkulmasta voi valita onko animaatio- vai riggaus-tilassa.

Tein pienimuotoisia kokeiluja ohjelmalla ja olin positiivisesti yllätynyt sen helppokäyttöisyydestä. Nimalla pystyy rakentamaan luuhierarkian, kiinnittämään IK kahvat, lisäämään meshin kuvaan ja tekemään skinnauksen. Aikaisempi kokemukseni animaatio-ohjelmista helpotti ohjelman käyttöä merkittävästi: tiesin mitä halusin tehdä ja ohjelmasta löysin työkalut vaivatta. Palojen nimet, hierarkia ja alkuperäisen lokaation saa suoraan tuottaa Photoshop-tiedostosta, mikä nopeuttaa työprosessia huomattavasti. Hahmon osat saa muutettua mesheiksi yhden napin painalluksella. Meshin voi piirtää itse tai sen voi luoda automaattisesti ja prosessi on samanlainen kuin Spinessä.

Pilvipalvelun avulla projekteja pystyy työstämään missä vain, kunhan pääsee kirjautumaan verkkoselaimella omalle tililleen: kokeilin Nimaa Ipad Pro-tabletillani ja onnistuin luomaan uuden projektin ja siirtämään siihen Photoshop-tiedostosta hahmon. Valitettavasti Nimalla en pystynyt tekemään enempää, sillä muut käyttöliittymän työkalut eivät reagoineet Ipad Pron kosketusnäyttöön. Toivottavasti Ipadilla työskentely on mahdollista tulevaisuudessa ohjelman kehittyessä, sillä Nima olisi ainoa kokeilemistani ohjelmista, jossa työskentelyä voisi tehdä muulla kuin PC:llä ja Macilla.



Kuvio 15. Hahmoa pystyy animoimaan myös ilman luita ja pelkillä meshin verteksien positiioilla. Kuvassa alhaalla oleva palkki on animaation aikajana.

Käyttäjänä koin Niman heikkouden olevan sen käyttöönoton ohjaamisen puuttumisessa: ensimmäiset videotutoriaalit ohjelmasta ilmestyivät vasta toukokuun 2017 alussa. Sain kutsun ohjelman rajattuun beta-testiin syksyllä 2016, en tiedä olisinko muuten löytänyt ohjelmaa itsenäisesti, sillä Nimasta löytyy edelleen hyvin vähän materiaalia ja käyttäjäkokemuksia netistä. Materiaalin puute on erityisesti hankalaa silloin, kun käyttäjällä on vaikeuksia ohjelman ja sen oppimisen kanssa ja ratkaisut näihin ongelmiin pitää keksiä itse. Live2D:n ja Spinen nettisivuilta löytyy paljon tietoa niiden toiminnoista, sekä tutoriaaleja, joita löytää myös artistien tekeminä. 2Dimensionsin sivuilla ei ainakaan projekti-

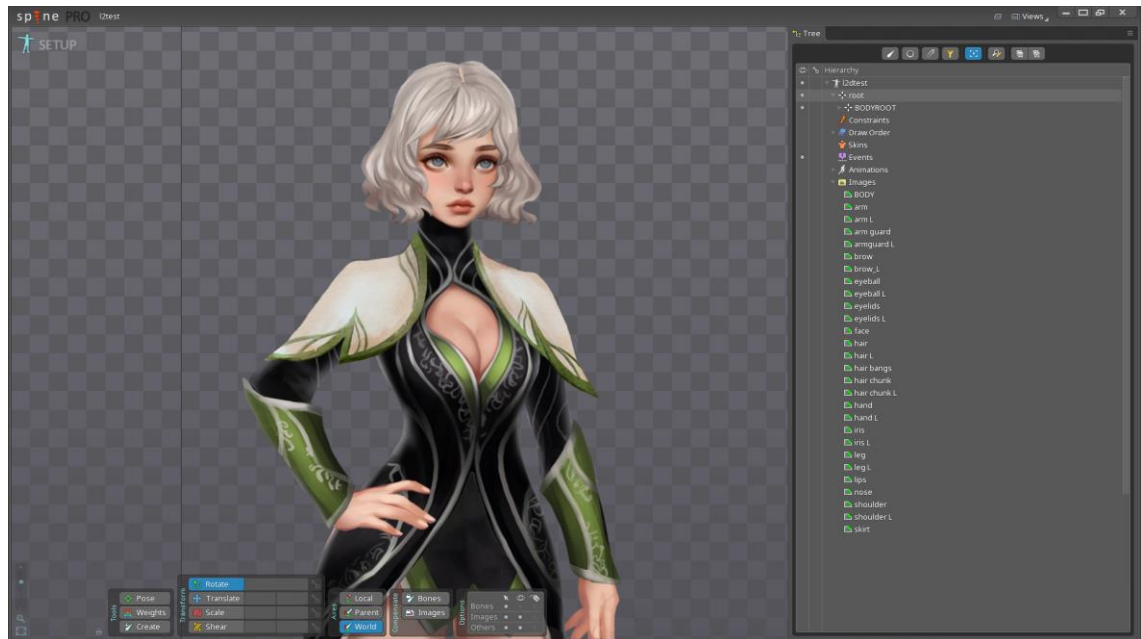
työtä suunnitellessani ollut julkaistu yhtään valmista peliä, jossa Nimaa olisi käytetty animaatioiden tekoon. Näiden syiden takia koin Niman olevan vielä liian kokeellinen ohjelma omaa projektityötäni varten, vaikka se vaikutti erittäin lupaavalta vaihtoehdolta Spinelle.

4.4 Riggaus ja animointi

Tämä vaihe on kahdessa eri osassa, jotta lukija voi helposti seurata työprosessin etenemistä eri ohjelmissa. En tule dokumentoimaan jokaista työvaihetta tai selittämään ohjelmien jokaista toimintoa vaan keskityn enemmän vaiheisiin, jossa käytetään mesh-defor-maatiota. Tämä koskee myös hahmon animointia. Pää tavoitteena on saada hahmosta animoitu kokonaisuus mahdollisimman lyhyessä ajassa. Olen käyttänyt Cubismia vain muutaman kerran aikaisemmin ennen tätä kokeilua ja olen paljon kokeneempi Spinen käyttäjä. Esitän Spinen workflown ensimmäisenä ja teen sillä vaativamman rigin, mutta perehdyn enemmän Cubismin käyttämiseen. Hahmon tavoitteiden lyhyt kertaus: Hahmolla pitää olla kasvoissa rigi, jolla voi tehdä useita eri ilmeitä, kehon ja kasvojen pitää kääntyä hieman, jotta sille voi luoda 2.5D-effektin.

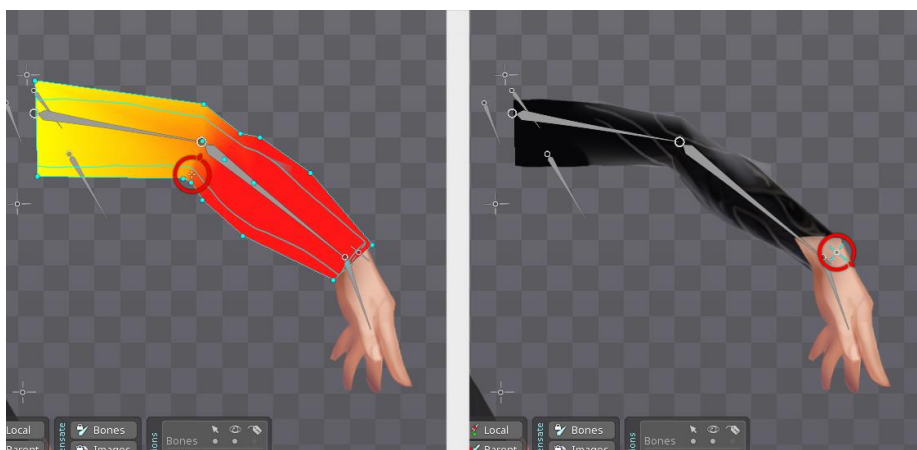
4.4.1 Spine Pro

Spineen pystyy tuomaan hahmon grafiikat nopeasti suoraan Photoshopista käyttäen valmista scriptiä (*script*), jonka löytää Spinen tiedostoista. Scripti vedetään avattuun Photoshopin tiedostoon ja sen avulla Photoshop exporttaa hahmon joka läyristä png-kuvan ja luo Photoshop-tiedostosta Spinen käyttämän JSON.-tiedoston. Scriptin etu on siinä, että kun JSON.-tiedoston avaa Spinessä, hahmon jokainen pala on oikeassa paikassa, palat on nimetty Photoshop-layereiden mukaan ja palojen kuvahierarkia eli kuvien näkyyjärjestys on myös suoraan Photoshopin layereistä.



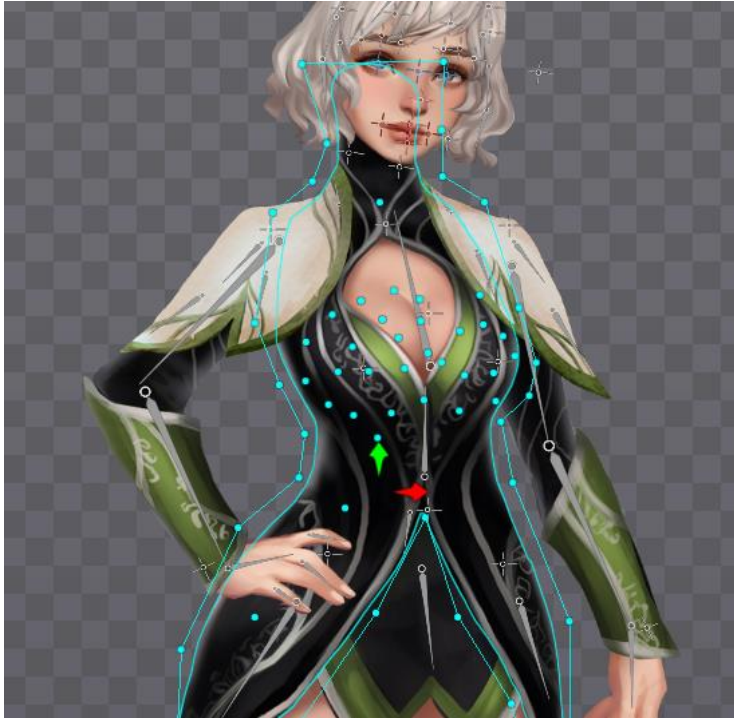
Kuvio 16. Exportattu hahmo Spinen käyttöliittymässä. Vasemmasta ylänurkasta voi vaihtaa animaation ja setupin (*setup*), eli riggauksen työtilat. Oikealla on hierarkia ikkuna, jossa hahmon kaikki osat on listattuna omissa kategorioissaan.

Hahmon riggaus alkaa luiden luomisella. Hahmon irralliset palat kiinnitetään luihin, jotta niitä pystyisi helposti animoimaan. Meshejä pystyy animoimaan joko sen verteksien muutoksilla tai siihen voi skinnata (*skinning*) kiinni luut. Tekniikka on samanlainen kuin 3D-grafiikan skinnauksessa. Skinnauksessa verteksien värit merkitsevät niiden vaikutusalueen. Tuhlasin liiankin paljon aikaa hahmon kehon rigin tekemiseen, mutta halusin kokeilla kehon kääntymistä ja käsien taipumista deformaation avulla. En käyttänyt IK-kahvoja, koska käsien liike oli niin vähäistä.



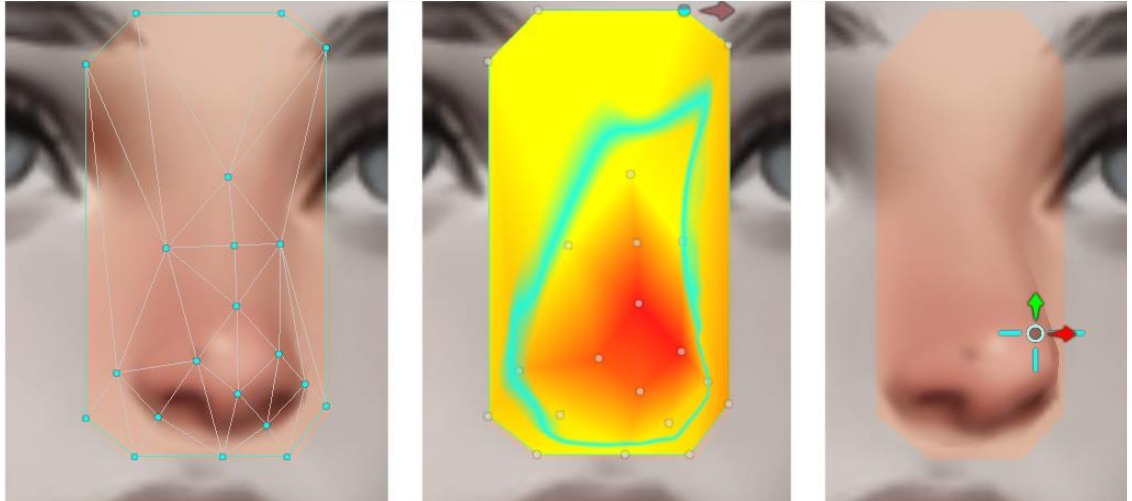
Kuvio 17. Skinnauksella ja luilla saa suorana maalattua kättä deformoitua, ilman sen pilkkomista palasiksi.

En kuvitellut kehon kääntymisen olevan niin haastavaa, mutta kokeilemalla opitaan. Meshin kuvassa oli niin paljon detaileja, jotka vääristyivät helposti ja niiden muoto oli vaikea pitää samana deformaatioissa. Varsinkin rintojen muoto muuttui nopeasti palikka-maiseksi. Lisäsin meshiin verteksejä, mikä auttoi hieman, mutta ongelman voi silti havaita valmiissa animaatioissa. Hahmon hengityksen animaation tein nopeasti yhden selkärangan luuta vähän skaalaamalla.



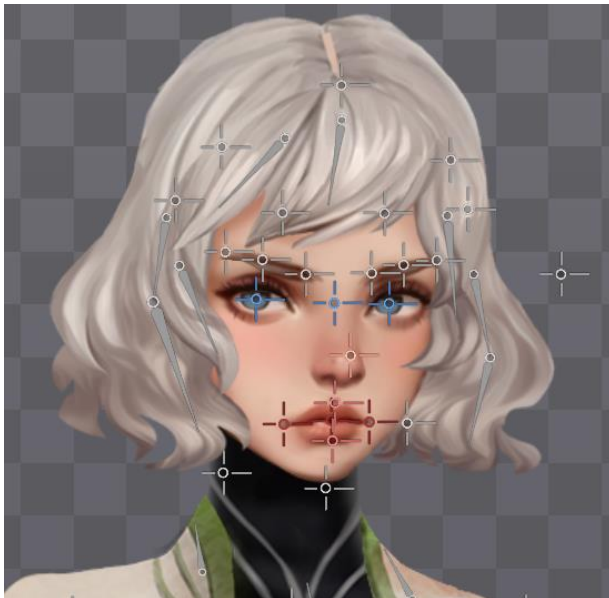
Kuvio 18. Hahmon kehon verteksejä muokaten sain luotua illusion kehon kääntymisestä. Kokeilin tätä vaihetta liian aikaisin ja palasin siihen taas vasta kun olin saanut hahmon rigin muuten valmiiksi, pystyin käyttämään tähän animaation enemmän aikaa, kun ajoitin sen viimeiseksi.

Kun olin saanut keholle siedettävästi toimivan rigin ja animaatiot, käytin lopun ajan hahmon kasvojen riggaamiseen. Silmät, kulmakarvat, nenä ja suu olivat kaikki meshejä, joissa oli kiinni useampi luu. Käytin luita meshien animoimiseen kaikissa muissa osissa paitsi hahmon silmissä. Myös hiukset skinnasin aluksi luihin, mutta sain tehtyä pelkällä verteksi pohjaisella mesh-deformaatiolla paljon uskottavamman animaation.



Kuvio 19. Hahmon nenän polygon mesh, jonka verteksit asettelin itse. Jotta sain luun kiinni meshiin niin, että vain nenän pääty animoituu sen mukana, laitoin sen vaikutuksen kohdistuvan vain punaisella merkittyyn alueeseen.

Moniosaista rigiä on hidasta animoida. Hieman työn helpottamiseksi laitoin iiriksille kontrollin, jolla niitä pystyi animoimaan samansuuntaisesti yhtä aikaan. Tein samoin kasvon muille osille, jotta voisin helpommin luoda illuusion hahmon pään kääntymisestä. Kontrollin käyttäminen ei kuitenkaan toiminut yhtä hyvin kuin kuvittelin, joten en tullut sitä tarvinneeksi.



Kuvio 20. Valmis kasvojen rigi, jonka nenän, silmän iiriksen ja suun luut on värikoodattu.



Kuvio 21. Naamalla pystyi tekemään toisistaan poikkeavia ilmeitä, vaikka huulien animaatio olikin rajoittunutta. Aukinaisen suun animoiminen olisi ollut liian työläistä kaiken muun lisäksi.

Silmien animaation tein frame-by-frame tyylisesti meshin verteksejä deformoiden. Olin positiivisesti yllättynyt lopputuloksesta. En odottanut meshin venymisen olevan niin vähäistä, että jos hahmo ei pitäisi silmiä kiinni kuin vain luomia räpytellessä, en näkisi niiden parantelua tärkeänä. Photoshopissa silmäluomelle olisi voinut maalata uuden version, jota olisi voinut käyttää luomien päällä silloin, kun hahmo pitäisi silmiä kauemmin kiinni.



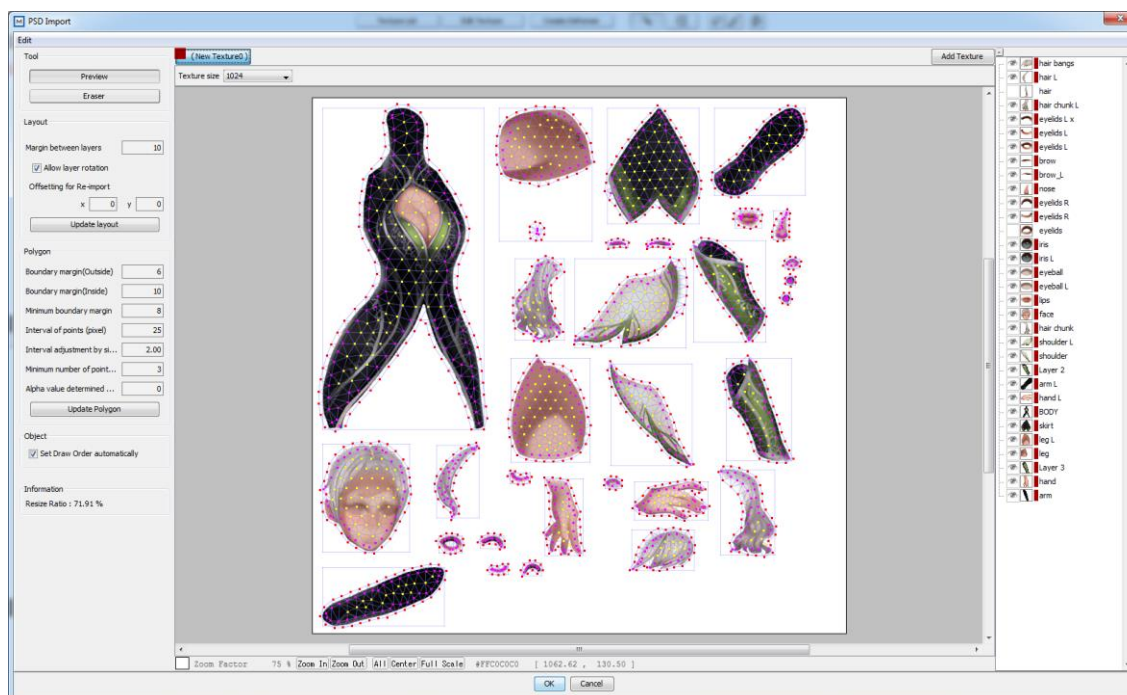
Kuvio 22. Silmien kolmen framen animaatio. Mesh-deformauksella pystyttiin korvaamaan yksittäiset frame-by-frame maalatut kuvat, jolla silmiä olisi voinut animoida.

Spinellä pystyy työstämään useampia animaatioita yhtä aikaa. Tein osan animaatiosta erikseen ja myöhemmin yhdistelin esimerkiksi hahmon kehon kääntymisen valmiiseen animaation. Lopuksi exporttasin valmiin animaation videoksi.

4.4.2 Cubism 2.1

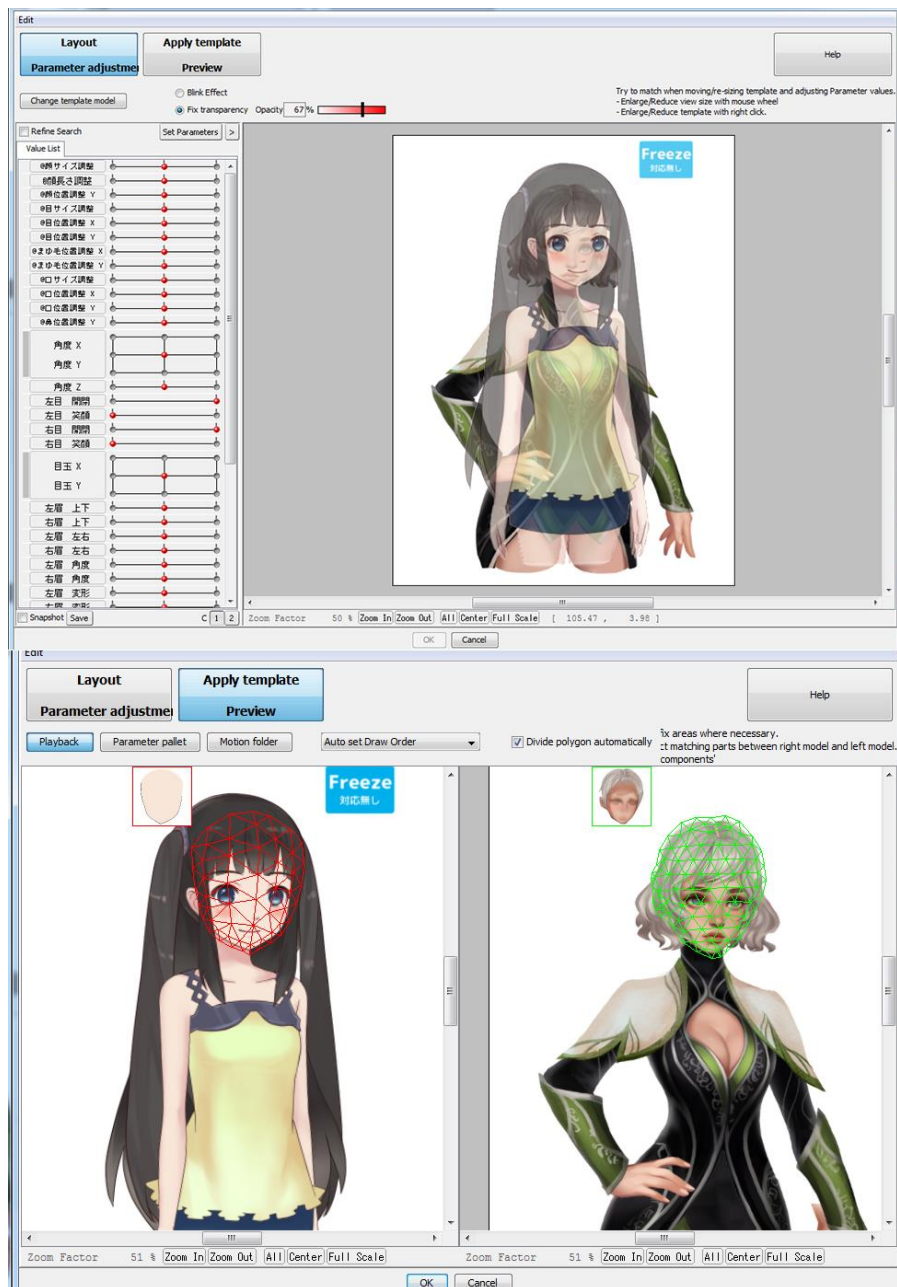
Cubismissa hahmon voi tuoda suoraan Photoshop-tiedostosta, yhdellä PNG-kuvalla, jossa on hahmon kaikki palat tai yksittäisinä kuvina. Photoshop-tiedoston tuominen avaa suoraan polygon mesh mapping -ikkunan, jossa osille luodaan polygon mesh. Meshin voi piirtää itse, tai ohjelma voi luoda sen automaattisesti joka osalle. Halusin kokeilla kuinka hyvin ohjelman tekemä mesh-ruudukko toimii ilman ongelmia, joten valitsin sen

vaihtoehdon. Pienen muokkauksen jälkeen hahmo oli nopeasti valmis seuraavaan vaiheeseen.



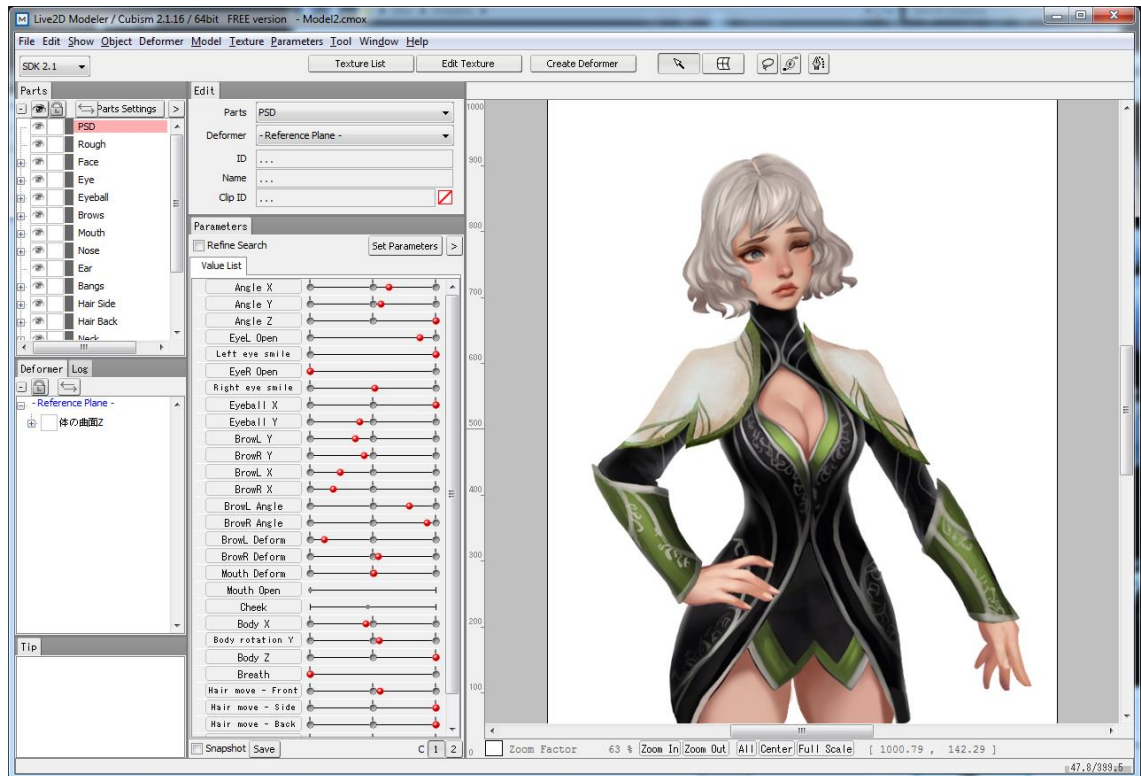
Kuvio 23. Mesh mapping –ikkunassa hahmon paloille on automaattisesti luoto mesh, jonka tiheyttä ja muita asetuksia voi muokata vasemmalla olevilla numeroilla. Oikealla on lista kaikista Photoshop-tiedostossa olevista kuvista. Kuvien nimen vieressä on pieni silmän kuva, josta näkee onko osa mukana mesh mappayksessä.

Cubismissa hahmon voi rigata alusta asti itse tai vaihtoehtoisesti voi käyttää ohjelman tarjoamia pohjia, joista saa hahmolle hyvän rigin mistä aloittaa. Ajan säästämiseksi päätin kokeilla valmista pohjaa hahmolleni. Pohjia oli useita erilaisia: kasvot olivat eri kulmissa ja niistä oli mies- ja naisversiot. Valitsin pohjan jonka kasvot olivat lähimpänä hahmoani. Pohjan ja hahmon yhdistämisessä pohjana toimivan kuvan mittasuhteita muokataan niin, että se sijoittuu mahdollisimman samankaltaisena hahmoni päälle. Tämän jälkeen pohjan osat yhdistetään hahmoni osiin. Kummatkin hahmot ovat animoituja ja pystyn helposti näkemään kuinka hyvin pohjan osat sopivat omaani ja mitkä osat vaativat vielä yhdistelyä. Olin yllättynyt kuinka hyvin pystyin käyttämään hahmoni osia pohjan avulla. Ainoa ongelmana oli silmien ylä- ja alaluomet, joita olin käyttänyt Spinessä samoina paloina. Jouduin menemään muokkaamaan Photoshop-tiedostoa ja leikkaamaan silmäluomet erillisiksi osiksi. Koko työn joutui aloittamaan alusta, mutta siinä ei mennyt kymmentä minuuttia kauempaa. Sain kaikki haluamani animaatiot suoraan pohjasta.



Kuvio 24. Pohjan käytössä on kaksi vaihetta. Ylemmässä ikkunassa tehdään mittasuhteiden muokkaus ja alemmassa ikkunassa hahmojen valitut osat yhdistetään toisiinsa.

Cubismin riggaus toimii siten, että hahmon polygon meshit kiinnitetään deformeriin. Nämä deformerit liitetään motion parametreihin.



Kuvio 25. Cubism 2.1:n käyttöliittymä. Vasemmalla on hierarkia, keskellä on motion parameteerit, joita manipuloimalla hahmon eri osia on deformoitu.

Deformereilla voi tehdä luurankomaisen hierarkian ja tämän hierarkian ylin deformer vaikuttaa koko kuvaan. Hahmoa pystyy deformata eri tavoin: rotator deformer kääntää sen alaisia kuvia, polygon mesh vaikuttaa suoraan tekstuuriin ja sen verteksejä voi siirrellä ja deformerilla voi muokata ja kontrolloida kokonaisuuksia. Deformer toimii hyvin samalla lailla, kuin Photoshopin warp mode –työkalu. Hahmo deformataan tiettyyn muotoon, tämä muoto tallennetaan motion parametreihin keyframiksi. Motion parametreihin voi tehdä useita keyframejä ja niiden tweenausta käytetään hahmon animoimiseen.



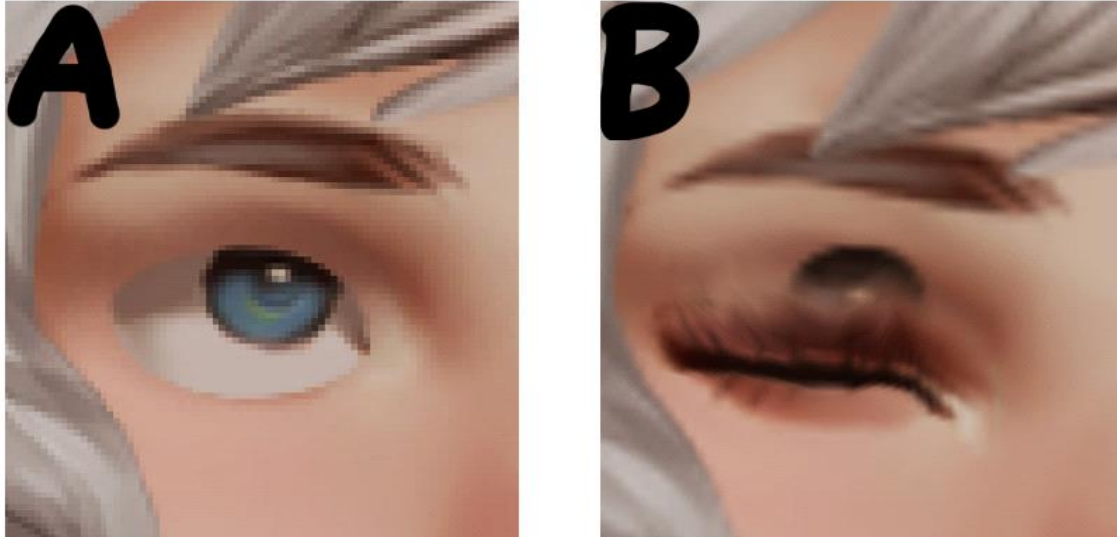
Kuvio 26. Vasemmalla on kehon polygon mesh, keskellä kädelle laitettu rotator deformer ja oikealla kehon deformer.

Pohjalla luodut animaatiot olivat karkeita ja käytin noin viisi tuntia hahmon rigin muokkaamiseen. Kun olin tyytyväinen hahmoni rigiin, exporttasin sen Cubism 2.1 Animatoriin. Animatorissa hahmoa animoidaan hahmon rigin motion parameterien kautta. Parametrien muokkaus animaation aikajanalla luo automaattisesti keyframien ja animaation eri deformaatiot sulautuvat toisiinsa pehmeästi. Keyframejen välistä animaatiota voi muokata halutessaan. Ilmaisversiossa animaation voi exporttaa vain PNG.-kuvasarjana tai gif.-animaationa.

Palasin myöhemmin ohjelmaan kokeilemaan eri asioita. Olisin halunnut kokeilla clipping maskia (*clipping mask*) hahmossa ja pettymyksekseni se ei ole ilmaisversiossa mahdollista. Koin sen kuitenkin niin mielenkiintoiseksi ja, että lisäsin siitä osion tähän opinnäytetyöhön. Clipping mask on artisteille tuttu työkalu Photoshopista ja Illustratorista. Clipping maskissa yksi valittu kuva toimii pohjana maskina (mask) siihen linkitetyille kuville ja se määrää niiden näkyvyyden. Linkitetyt kuvat eivät siis ole näkyvinä maskin ulkopuolella. (Adobe Inc. n.b.)

2D-kuvana silmän iiris on parempi pitää erikseen silmämunasta, jotta sitä pystyy animoimaan helpommin. Yksi silmiin liittyvä ongelma on se, että iiris ei pysy silmämunan sisällä ja esimerkiksi tulee näkyväksi luomen alta, jos sitä liikuttaa liikaa. Kuviossa 27. näkyy kuinka luomen mesh on deformoitu normaalia kapeammaksi ja iiris tulee näkyviin sen alta. Tässä tilanteessa clipping mask olisi ollut loistava ratkaisu ongelmaan. Valitettavasti Cubism 2.1. ilmaisessa versiossa clipping mask on lukittu, eikä sitä voi käyttää, mutta

Spinessä maskaamista ei ole ollenkaan. Cubismissa jouduin rajoittamaan iiriksen liikealueen pieneksi ja välttämään hahmon katsomista ylös samalla kun luomi on kiinni. Jos hahmoa olisi käytetty pelissä, olisin voinut maalata yläluomen erillisenä palana ja käyttää sitä iiriksen piilottamisessa.



Kuvio 27. A:lla merkityssä kuvassa silmän iiris on Photoshopissa maskattu silmän valkoiseen osaan. Silmäluomet on piilotettu kuvan selventämiseksi. Maski estää iiriksen liikkumasta sen ulkopuolelle. B:llä merkityssä kuvassa Cubismissa rigattu hahmo katsoo ylöspäin samalla kun luomi on suljettu.

5. Pohdinta

Spinellä ja Cubism 2.1 kummallakin pystyy tekemään haluamaani hahmo-animaatiota. Spinessä käytin riggaukseen yli 20 tuntia, josta suurin osa meni mesh-deformaation ja luuanimaation muokkailuun. Spinessä riggaaminen oli helppoa ja sain hahmon tekemään mitä halusin, mutta mitä yksityiskohtaisemmaksi rigi meni, sitä hitaampaa animoimisesta tuli. Tämä tuli erityisesti ilmi hahmon kasvoissa. Hahmon suun olisin voinut pystyä animoimaan pelkästään meshin vertekseillä tai vain kahdella kontrollerilla. Viisi kontrolleria hidastivat animaatiota, koska hiirellä klikatessa niistä oli vaikea valita oikea kontrolli ensimmäisellä yrityksellä. Sain kuitenkin hahmo rigistä juuri sellaisen kuin halusin ja tarvitsin. Rigi ei mennyt rikki ja animaatioat toimivat niin kuin niiden pitikin, mutta olisin voinut esitellä rigin ominaisuuksia paremmin animaatiossa.

Cubism 2.1 riggaukseen meni yhteensä noin viisi tuntia ja lyhyeen testianimaatioon noin puolituntia. Siedettävän rigin sain aikaan alle puolessa tunnissa käyttäen valmista pohjaa. Jos pohja olisi ollut hahmolleni sopivampi tai jos olisin suunnitellut hahmoni Cubismin pohja-malleja referoiden, olisi workflow Photoshop-tiedostosta rigin hiomiseen ollut vieläkin nopeampaa. Sopivalla pohjalla voisi kokeiluni perusteella tehdä useamman kuin yhden hahmon, joilla olisi sama rigi ja myös animaatio. Spinessä on skin-systeemi, jolla samalla luurangolla voi tehdä useamman hahmon vaihtamalla sen kuvat. Cubismin etuna on kuitenkin se, että sen pohjalla voi monistaa meshejä, jotka ovat kiinni isossa deformerissa, eli jos kuvan mesh hajoaa, deformerit toimii siitä huolimatta.

Taidettyli vaikutti hahmon toteutukseen merkittävästi niin maalaamisen kuin animoimisen kannalta ja tämä tuli myöhemmin esille varsinkin projektin hahmon kasvojen animoinnissa. Varsinkin nenä tuotti ongelmia, kun hahmon päätä käänsi. Nenä muuttui nopeasti epäluonnollisen näköiseksi ja irralliseksi jos sitä liikutti liikaa. Spinessä pään kääntäminen oli hyvin rajoitettua ja Cubismissa en saanut nenästä käyttökelpoista hahmon katsoessa ylös. Cubismissa ongelma oli luultavasti deformereissa ja ne saattoivat olla syynä nenän karkailuun kasvojen alueelta. Kehon kääntämisessä tuli ongelmia kummasakin ohjelmassa: Spinessä en saanut skinnattua tarpeeksi geometriaa hahmon rinnuksen alueelle, jossa oli paljon yksityiskohtia, mikä aiheutti kuvan ei-haluttua vääristymistä. Cubismissa huomasin myöhemmin, että verteksi-pohjaisen deformaation sijaan deformerien käyttäminen antoi paljon luontevamman lopputuloksen ja hahmon kääntäminen oli paljon helpompaa Cubismissa kuin Spinessä.

Cubismilla animaatiota pystyi testaamaan riggaus-vaiheessa motion parametreita liikuttaessa, mutta minulla kävi vahingossa monesti niin, että joku parameter oli jäänyt päälle ja olin lähtenyt muokkaamaan toista deformausta. Virheen huomasin vasta myöhemmin, kun esimerkiksi kasvot lähtivätkin aivan eri suuntiin kuin aikaisemmin. Cubismin animaattori hajosi, kun olin yrittämässä saada animaatiostani ulos gif.-animaatiota ja en saanut sitä rendattua ulos. Tiedostosta katosi kaikki grafiikka, enkä löytänyt ohjelmasta mitään keinoa, jolla sen olisi saanut takaisin. Suurin osa Live2D:n dokumenteista on japaniksi, joten en löytänyt mistään apua ongelmaani. Tämän takia opinnäytetyön liitteissä on vain Spinellä tehdyn animaatio.

Seuraavaa vastaavanlaista projektia varten valitsisin Cubismin, Animatorin ongelmista huolimatta. Deformereilla sai aikaan isoja muutoksia ja hahmolle sai nopeasti parametrejä manipuloimalla monta eri osaa liikkumaan yhtä aikaa. Hahmon pystyi kääntämään deformereilla luonnollisen näköisesti ja pohjakuvan kautta riggaaminen oli nopeaa. Cubism osoittautui yllättävän laadukkaaksi ohjelmaksi vaikka käytössäni oli vain sen rajoitettu ilmaisversio. Aion tulevaisuudessa keskittyä Live2D:n ohjelmien opetteluun ja ohjelmasta on jo uusi versio olemassa. Cubism 3, jossa on varmasti paranneltu monia vanhemman ohjelman ongelmia. Jos hahmolleni pitäisi tehdä paljon suuriliikkeisempiä animaatioita, esimerkiksi Dragons' Crown tyylistä peliä varten, valitsisin siihen tarkoitukseen Cubismin sijaan Spinen. Koin Spinen vahvuuksien olevan sen riggauksessa: luurangon luominen oli helppoa ja ohjelmassa pystyi tekemään animaatiota ja muokkaamaan rigiä samaan aikaan, toisin kuin Cubismissa. Spinellä pystyi helposti tekemään frame-by-frame animaatiota keyframejen kautta, kuten silmien räpyttämisen. Frame-by-frame oli mahdollista Cubismissa, mutta sitä varten joutui rigin exporttaamaan toiseen ohjelmaan.

Opinnäytetyön tekeminen opetti minulle tarpeellisista työvaiheista, joita 2D-deformaatioanimaatio vaatii, varsinkin turhien vaiheiden karsimisen tärkeydestä. Dragon's Crownin hahmojen animaatioiden analysoinnin sekä projektityön tekemisen jälkeen koen uskoni 2D-deformaatioanimaatioon tehokkuuteen peligrafiikassa olevan entistä vankempi. Pystyn argumentoimaan 2D- ja 2.5D-grafiikan puolesta ja perustelemaan miksi mesh-deformaatio ja luurankoanimaatio on sen animoimiseen loistava ratkaisu. Projektityöni jälkeen pystyn valitsemaan tulevaisuuden projektien ohjelmat helpommin. Tulevaisuudessa osaan myös paremmin välttää toistamasta samoja virheitä, joita esimerkiksi hahmon kasvojen rigissä ilmeni ja seuraavissa projekteissa pystyn käyttämään aikaa enemmän uusien asioiden tekemiseen ja kuin aikaisempien tekemisten korjaamiseen ja paranteluun.

Lähteet

Adobe Inc. n.d. Clipping Masks

<<https://helpx.adobe.com/photoshop-elements/using/clipping-masks.html>> (luettu 1.5.2017)

Adrien-Luc, Sanders n.d. What is 2.5D Animation?

<<http://animation.about.com/od/faqs/f/What-Is-2-5d-Animation.htm>> (luettu 20.2.2017)

Aim n.d. Animation Notes #6 Pose to Pose Animation

<http://minyos.its.rmit.edu.au/aim/a_notes/anim_pose.html> (luettu 1.5.2017)

Autodesk Maya 2015a. Introduction to polygons

<<https://knowledge.autodesk.com/support/maya-lt/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/MayaLT/files/Polygons-overview-Introduction-to-polygons-htm.html>> (luettu 4.3.2017)

Autodesk Maya 2015b. Introduction to UV mapping

<<https://knowledge.autodesk.com/support/maya-lt/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/MayaLT/files/GUID-50A2C6CD-C1D0-40DB-909F-90D163CD4886-htm.html>> (luettu 4.3.2017)

Benson, James 2015. Youtube. GDC: The Animation Process Of Ori & The Blind Forest

<<https://youtu.be/m8IOwrWNbEY>> (katsottu 2.2.2017)

Bloop Animation, n.d. 5 Types of Animation

<<https://www.bloopanimation.com/types-of-animation/>> (luettu 4.3.2017)

Codrina 2016. Advantages and disadvantages of skeletal versus sprite sheet animations

<<http://marionettestudio.com/advantages-and-disadvantages-of-skeletal-versus-sprite-sheet-animations/>> (luettu 25.4.2017)

Dooby, Evan 2014. Frame vs. Mesh Animations

<<http://wulverblade.com/frame-vs-mesh-animations/>> (luettu 20.2.2017)

Dragon's Crown Wikia. n.d.a Fighter Sprite

<<http://dragons-crown.wikia.com/wiki/Category:Sprites>> (katsottu 2.5.2017)

Dragon's Crown Wikia. n.d.b Sorceress Sprite Sheet

<http://dragons-crown.wikia.com/wiki/File:DC_si.png> (katsottu 2.5.2017)

Esoteric Software. n.d. Blog

<https://esotericsoftware.com/blog> (luettu 3.3.2017)

Ikinamo 2014. Youtube. Live2D Euclid animates 2D illustrations in rich 3D environments #DigInfo

<<https://youtu.be/8SMDLnC-cMU>> (katsottu 3.3.2017)

Lambert, Steven 2013. An Introduction to Sprite sheet Animation
<<https://gamedevelopment.tutsplus.com/tutorials/an-introduction-to-spritesheet-animation--gamedev-13099>> (luettu 23.4.2017)

Live2D Inc, n.d.a About
<<http://www.live2d.com/en/about>> (luettu 20.2.2017)

Live2D Inc. n.d.b Cubism Documents
<http://sites.cybernoids.jp/cubism_e/> (luettu 20.2.2017)

Rosso, Guido. n.d. Welcome to 2D
<<https://www.2dimensions.com/b/53-welcome-to-2d>> (luettu 2.5.2017)

Sheffield, Brandon n.d. King of 2D: Vanillaware's George Kamitani
<http://www.gamasutra.com/view/feature/132486/king_of_2d_vanillaware_george_.php?print=1> (luettu 3.3.2017)

The Spitters Resource. n.d. Dragon's Crown - Miscellaneous - Character Portraits
<<https://www.spritters-resource.com/fullview/57242/>> (katsottu 1.5.2017)

Vanillaware. n.d. Dragon's Crown Red Dragon
<<http://dragons-crown.com/#gallery>> (katsottu 2.5.2017)

Wikipedia n.d. Dragon's Crown
<https://en.wikipedia.org/wiki/File:Dragon%27s_Crown.jpg> (katsottu 10.5.2017)

Wikipedia 2017. UV wrapping
<https://en.wikipedia.org/wiki/UV_mapping> (luettu 4.3.)

Wisfire. n.d. Herald: An Interactive Period Drama
<<http://www.live2d.com/en/interviews/herald>> (luettu 5.3.2017)

Verkkolinkki Spinellä tehtyyn animaatioon

<https://vimeo.com/219152034>