



# Perinteisten maalaustekniikoiden ja 3D-grafiikan yhdistäminen 2D-pelin luomisessa

Hannu Timonen

OPINNÄYTETYÖ  
Tammikuu 2022

Tietojenkäsittelyn tutkinto-ohjelma  
Pelituotanto

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Pelituotanto

TIMONEN, HANNU:

Perinteisten maalaustekniikoiden ja 3D-grafiikan yhdistäminen 2D-pelin luomisessa

Opinnäytetyö 47 sivua, joista liitteitä 0 sivua  
Maaliskuu 2022

---

Opinnäytetyössä luotiin 2D-mobiilipeli OWLS, jonka avulla tutkittiin perinteisten maalaustekniikoiden ja 3D-grafiikan hyödyntämistä 2D-pelissä. Työssä toteutettiin sekä peliympäristöjä, pelihahmoja että pelihahmojen animaatioita.

Työn tavoitteena oli tutkia edellä mainittujen tekniikoiden soveltuvuutta 2D-pelin tuotantoon ja selvittää artistin työnkulkuja näitä tekniikoita käyttäessä. Peliin luotiin lukuisia peliympäristöjä ja pelihahmoja käyttäen maalaus-, 2D- ja 3D-tekniikoita ja tarkasteltiin parhaita tapoja käyttää niitä tunnelman ja kauniin peligrafiikan luomiseen.

Opinnäytetyössä pohdittiin perinteisten maalaustekniikoiden käyttöä peligrafiikassa käyttäen esimerkkejä pelihistoriasta sekä työtä varten luodusta mobiilipeleistä. Työssä tuotiin esiin myös grafiikan luomisen perusasioita, ja niiden kautta pyrittiin antamaan lukijalle käsitys 2D- ja 3D-grafiikkaan vaadittavista taidoista. Opinnäytetyössä käsiteltiin artistin näkökulmasta myös peliympäristöjen ja hahmojen valaistukseen liittyviä lainalaisuuksia. Työssä sivuttiin animaatioita ja efektointia, mutta pääpaino työssä oli grafiikan luomisessa.

Työn aikana havaittiin, että käyttäen perinteisillä maalaustekniikoilla ja 3D-mallinnuksella, on mahdollista luoda johdonmukaista ja kaunista grafiikkaa. Jos monimediatekniikka toteutetaan huolellisesti, sillä voidaan luoda pelille vahva visuaalinen identiteetti. Maalaustekniikoilla ja 3D-grafiikkaa teksturoimalla voidaan kaksiuulotteisen pelin ulkoasusta saada yhtenäinen ja toimiva.

---

Asiasanat: 3D-grafiikka, 2D-grafiikka, taide, mobiilipeli, peligrafiikka

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Bachelor of Business Administration  
Game Production

TIMONEN, HANNU:  
Combining Traditional Art and 3D graphics in 2D Game Design

Bachelor's thesis 47 pages, appendices 0 pages  
March 2022

---

A 2D mobile game OWLS was created for the purposes of this bachelor's thesis. The mobile game was used as an example to study how combining traditional art techniques and 3D-modeling work when creating a 2-dimensional game. Multiple game environments, characters and animations were created for the thesis.

The objective of this Bachelor's thesis was to review the suitability of traditional art techniques and 3D modelling techniques in 2D game development, and study the workflow of an artist using these methods. The best possible work flows for creating game environments, characters, and animations were studied in order to create atmosphere and beautiful graphics.

The use of traditional art was reviewed in the context of games through historical examples, as well as examples from OWLS mobile game. The basic principles of graphic design were introduced to give a wide perspective of the skills needed to create 2D and 3D graphics. The basic principles of lighting environments and characters were covered briefly in the thesis. Animation and effects were introduced, but game environments and character creation were the main focus.

The thesis shows that combining traditional art and 3D-graphics can result in cohesive and beautiful graphic design for games. When executed properly, mixed media technique is a viable way to create a strong visual identity for a game. With the use of painting methods, combined with the use of textures in 3D-models, it is possible to achieve a solid game layout.

---

Key words: 3D graphics, 2D graphics, art, mobile game, game graphics

## SISÄLLYS

1	JOHDANTO .....	5
2	PELIYMPÄRISTÖJEN LUOMINEN .....	6
2.1	Yleisesti peliympäristöistä .....	6
2.2	Käsin maalatut ympäristöt .....	9
2.2.1	Maalausprosessi .....	12
2.2.2	Maalauksesta digitaaliseksi taustaksi .....	14
2.3	Digitaalisesti luodut ympäristöt .....	19
2.3.1	3D-mallit ja mallien renderointi .....	20
2.3.2	Renderoinneista taustaspriteiksi .....	27
2.3.3	Parallax scrolling .....	30
3	PELIHAHMOJEN LUOMINEN JA ANIMOINTI .....	32
3.1	Pelihahmojen luominen .....	32
3.2	Pelihahmojen animaatiot .....	38
3.2.1	Riggaus .....	38
3.2.2	Animointi .....	41
3.3	3D-animaatiot sprite-animaatioiksi .....	42
4	EFEKTIT .....	44
4.1	Efektointi osana peliympäristöä .....	44
5	POHDINTA .....	45
	LÄHTEET .....	46

## 1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on luoda lukijalle selkeä kuva grafiikan, hahmojen ja peliympäristöjen luomisesta 2D-mobiilipeliin. Työ ei ole opas 2D- tai 3D-grafiikkaohjelmistoihin vaan se pohtii artistin työkulkua valmiiden peligrafiikoiden aikaan saamiseksi. Työ tarkastelee ennen kaikkea, miten saadaan yhdistettyä 3D-grafiikkaa ja perinteisiä maalaustapoja ja näiden avulla luotua kiinnostavaa ja johdonmukaista peligrafiikkaa kaksiulotteiseen peliin. Työtä varten on luotu 2D-mobiilipeli OWLS, jonka ratkaisujen kautta tuodaan esiin työtapoja ja ongelmakohtia monimediagrafiikan tuottamisessa peleihin. Työssä käydään läpi peliympäristöjen luomista sekä käsin maalaamalla että digitaalisten ohjelmistojen avulla, käydään läpi pelihahmojen luomista ja animoimista sekä erilaisten efektien merkitystä lopputuotteen tunnelmaan. Lukijalle tulee selkeä käsitys 2D- ja 3D-ohjelmistojen hyödyntämisestä ja työvaiheista pelituotannossa artistin näkökulmasta. Esimerkeissä on käytetty Adobe Photoshopia, Blenderiä ja Unitya, mutta opinnäytetyössä kuvatut asiat eivät ole ohjelmasidonnaisia vaan antavat ennemminkin suuntaviivoja peliartistin työhön. Työssä käydään läpi myös tunnettuja esimerkkejä peleistä, joissa on käytetty perinteisiä maalaustekniikoita halutun graafisen ilmeen saavuttamiseksi. Näistä esimerkeistä saatuja ajatuksia ja inspiraatiota on myös hyödynnetty opinnäytetyötä varten tehdyn OWLS-pelin ulkoasun suunnittelussa. Työ antaa peliartisteille ideoita peligrafiikan toteutukseen ja pyrkii korostamaan artistin työn monipuolisia mahdollisuuksia.

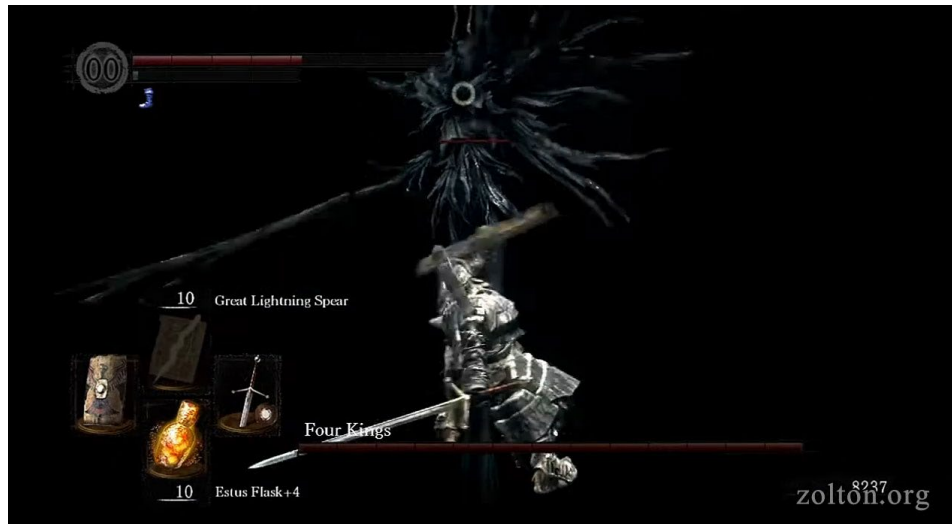
## 2 PELIYMPÄRISTÖJEN LUOMINEN

### 2.1 Yleisesti peliympäristöistä

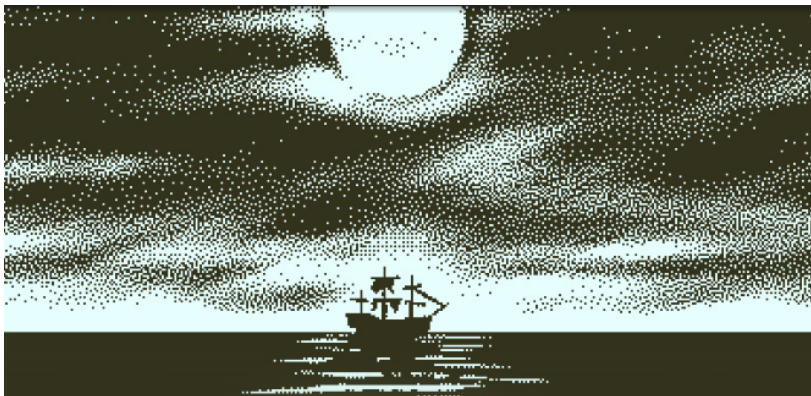
Peliympäristöllä tarkoitetaan tässä opinnäytetyössä nimenomaisesti pelin taustagrafiikoita. Peliympäristö on mielestäni parempi termi, sillä niin sanotut taustagrafiikat voivat usein olla myös pelinäköymän etualalla ja siksi taustagrafiikka ei mielestäni ole kuvaava termi. Kauneus on myös sana, joka työssä esiintyy useammassa kohdassa. Kauneus on tietenkin suhteellinen käsite, mutta tämän työn viitekehysessä se tarkoittaa hyvin toteutettua ja toimivaa grafiikkaa, joka antaa pelille huolitellun ja aiheeseen sopivan ulkoasun.

Peliympäristö on oleellinen osa peliä, joka luo pelille tunnelman ja kontekstin. Ympäristöt voi jakaa karkeasti neljään eri kategoriaan: Yksivärinen tausta, kuvatausta, animoitu tausta sekä vierivä tausta. (Morrison 2002, 159–160) Tässä työssä tarkastellaan erikseen myös vierivän taustan erikoistapausta, parallax scrollingia. Kuvissa 1–3 on esitetty esimerkit eri taustatyypeistä.

**Yksivärinen tausta** ei varsinkaan nykypäivänä ole enää laajalti käytössä. Se on lähtökohtaisesti tylsin ratkaisu eikä luo pelin tapahtumille minkäänlaista viitekehystä. Joitain poikkeuksia kuitenkin löytyy. FromSoftware'n kehittämässä ja Namco Bandai Gamesin vuonna 2011 julkaisemassa pelissä *Dark Souls* on eräässä taistelussa käytetty täysin mustaa ympäristöä. Koska kyseessä on kolmiulotteinen peli, mustan taustan käyttö vie pelaajalta etäisyyden- ja paikantajun ja toimii näin eräänlaisena pelillisenä elementtinä. Se myös poikkeuksellisesti asettaa kontekstin, sillä se kuvaa kadotusta, tyhjyyttä.



**Kuvatausta** on jo huomattavasti yksiväristä taustaa mielenkiintoisempi. Kuvataustojen ongelma on kuitenkin liikkumattomuus. Koska kuva on staattinen, mikään ei kerro pelaajalle sen olevan todellinen vaan se näyttää lähinnä taustakan-kaalta. (Morrison 2002, 160) Kuvatausta on kuitenkin paremman puutteessa riit-tävän hyvä vaihtoehto ja tarjoaa pelaajalle sekä visuaalisuutta että viitekehysten.



**Animoitu tausta** tuo kuvataustaan verrattuna peliin elävyyttä ja uskottavuutta. Animoitussa taustassa esimerkiksi oksat voivat heilua tai tähdet tuikkia. Morrison (2002, 160) kuitenkin muistuttaa, että koska taustakuvat ovat melko suuria, ei

koko kuvaa voi jatkuvasti piirtää uudestaan animaation saavuttamiseksi. On kuitenkin mahdollista käyttää elävyyden saavuttamiseksi muita metodeja. Taustan päälle voi tehdä pienempiä liikkuvia elementtejä, jotka sulautuvat taustaan, mutta eivät ole osa itse taustakuvaa. On myös mahdollista ohjelmallisesti esimerkiksi vaihtaa taustakuvan valoisuutta tai sävyä, jolloin voidaan saada aikaan vaikkapa tunnelma takkatulen loimusta.



KUVA 3. Esimerkki animoidusta taustasta. Kuvassa näkyvä vesiputous on animoitu. Kuvakaappaus pelistä Ori and the Will of the Wisps. (2020)

**Vierivä tausta** on paljon ruutua suurempi kuva tai kokoelma graafisia elementtejä, joita liikutellaan ohjelmallisesti ruudun poikki (Morrison 2002, 160). Vieriviä taustoja voidaan käyttää monella tavalla. Tausta voi liikkua pelihahmon mukana, jolloin saadaan luotua tunne hahmon etenemisestä maailmassa. Voidaan myös animoida pelihahmo liikkumaan paikallaan ja vierittää taustaa tasaisesti pois päin pelaajan kuvitellusta liikesuunnasta, jolloin saadaan aikaan illuusio etenevästä liikkeestä. Vierivien taustojen haasteena on, että taustakuvista voi tulla hyvinkin

suuria. Tämä on toki vältettävissä niin, että kuvat pilkotaan toisiinsa saumattomasti sopiviin osiin ja niitä ladataan pelin aikana toistensa perään, jolloin kuvakoko pystytään pitämään järkevänä. Yksi tapa on myös tehdä yksi ainoa kuva, jota toistetaan yhä uudestaan. Tällöin kuvan suunnittelussa pitää käyttää erityistä tarkkuutta, jotta kuvan vasen ja oikea reuna sopivat yhteen eikä pelaajalle tule näkyviin saumakohtaa.

**Parallax scrolling** on vierivän taustan erikoistapaus. Tällöin vieriviä taustoja on useita päällekkäin. Jotkut näistä voivat olla myös pelihahmojen etupuolella. Kun taustoja vieritetään yhtä aikaa, mutta eri nopeuksilla, saadaan luotua illuusio kolmiulotteisesta pelimaailmasta. Tämä mahdollistaa todella näyttävien taustojen luomisen ilman oikeaa 3D-pelimaailmaa. Parallax scrollingiin palataan vielä tarkemmin myöhemmin tässä työssä.

Peliympäristöjen suunnittelua voidaan hyvin verrata elokuvien lavastamiseen. Jotta voidaan luoda toimivia ja peliin sopivia ympäristöjä, täytyy ymmärtää pelin maailma. Ympäristöjä tehdessä on tärkeää ymmärtää muun muassa pelin tarina, teema ja genre. Myös maailman koko saattaa vaikuttaa päätöksiin suunnittelussa. Usein peliprojekteissa ympäristösuunnittelu on oma työtehtävänsä. Suunnittelija työskentelee art directorin ja lead artistien kanssa, jotta maailmasta saadaan halutun tuntuinen. (Ahearn 2008, xvi–xvii)

## 2.2 Käsien maalatut ympäristöt

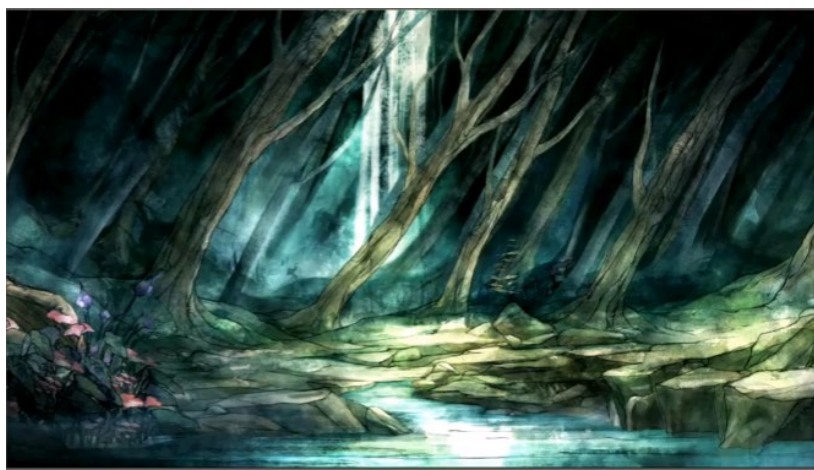
Työssä tarkastellaan, kuinka perinteiset maalaustekniikat toimivat osana digitaalisen pelin peliympäristöjä. Maalaus- ja piirustustekniikoita käytetään melko vähän nykypelien lopullisissa grafiikoissa ja näin ollen se kaipaa lisätarkastelua. Peleihin toki tehdään konseptikuvituksia, jotka toimivat pohjana varsinaisille grafiikoille. Glyn Williams (2016) toteaa, että on huomattavasti halvempaa tehdä esimerkiksi hahmoista konseptikuvia perinteisin menetelmin kuin käyttää aikaa 3D-mallien tekemiseen heti alussa. Rahaa säästyy, kun ideoita saadaan nopeasti koko tiimin nähtäville. Tavallisesti näitä kuvituksia käytetään kuitenkin lähinnä

idean myymiseen muulle tiimille. (Williams 2016) Onkin siis kiinnostavaa tarkastella, olisiko näitä käsin maalattuja kuvituksia mahdollista käyttää myös valmiissa pelituotteessa. Tämä voisi antaa kaupallisissa projekteissa mahdollisuuden saavuttaa upeita graafisia elementtejä pienemmillä kustannuksilla. Maalauksien ja ylipäänsä 2D-grafiikan etu on myös tietynlainen yksinkertaisuus 3D-grafiikkaan verattuna. Clint Cearley (2020) kertoo opetusvideossaan juuri tästä erosta. 2D-grafiikka perustuu havainnointiin eikä kuvittajan tarvitse välttämättä ymmärtää, miten kappale muodostuu, miten se toimii tai minkälaista materiaalia se on. 2D-grafiikka perustuu lähinnä havainnoidun asian toisintamiseen yksinkertaisempaan. 3D-grafiikassa sen sijaan on tärkeää ymmärtää kappaleen ominaisuudet. 3D-grafiikkaa tehdessä täytyy myös käyttää mielikuvitusta, ellei referenssikuvia ole saatavissa, sillä kappale ja sen mittasuhteet täytyy ymmärtää joka kuvakulmasta. (Cearley 2020)

Monet kuvankäsittelyohjelmat ja digitaaliset maalausohjelmat ovat syrjäyttäneet perinteisen maalauksen pelialalla. Eikä vanhoissakaan peleissä maalausjälkeä voitu nähdä valmiissa pelissä. Tämä johtui yksinkertaisesti teknisistä rajoituksista. 80-luvun lopun ja 90-luvun alun grafiikkakortit pystyivät toistamaan korkeintaan 256 eri väriä ja pelikoneiden prosessoritkin olivat 8- tai parhaimmillaan 16-bittisiä. (Wolf 2007) Edellä mainituilla teknisillä ominaisuuksilla ei yksinkertaisesti voitu saavuttaa lopputulosta, joka nykytekniikalla on mahdollista. Nykyaikaiset piirtopöydät ja ohjelmistot mahdollistavat myös erinomaisesti perinteisen maalauksen tai piirtojärljen imitoinnin. Digitaalisen grafiikan suuria etuja verrattuna perinteiseen ovat myös nopeus ja virheiden korjaamisen helppous. On myös muistettava, ettei alkuperäistä maalausjälkeä voi koskaan täydellisesti tuoda tietokonepeliin, sillä muun muassa pensselinjäljet, paperin korkeuserot ja maalin korkeuserot eivät voi toistua sellaisenaan ainakaan 2D-pelissä.

LucasArtsin 90-luvun alussa julkaisemien *The Secret of the Monkey Island 2: LeChuck's Revengen* ja *Day of the Tentacle*n artisti Peter Chan kertoo, että kahden edellä mainitun pelin välillä hän joutui vaihtamaan maalaustekniikkaansa täysin. Hän huomasi, että 90-luvun alun digitaalinen grafiikka rajoitti lopputulosta ja käytännössä tuhosi hänen maalausjälkensä, koska sitä ei tietenkään voinut sellaisenaan siirtää peliin. *Day of the Tentacle*a varten hän muutti ja yksinkertaisti tyyliään niin, että digitaalinen versio olisi mahdollisimman lähellä sitä, mitä hän itse oli maalannut. (DoubleFineProd, 2016)

Perinteisiä maalaustekniikoita on kuitenkin käytetty peleissä myös sellaisenaan. Yksi tunnetuimpia esimerkkejä pelistä, jossa käsinmaalatut konseptimaalaukset ovat suoraan nähtävissä pelissä on Ubisoftin 2014 julkaisema *Child of Light*. Pelin ohjaaja Patrick Plourde (2014) kertoo, että tekijät halusivat pelin tuntuvan kuin olisi maalauksen sisällä. Vesivärein maalattua konseptitaidetta käytettiin suoraan auttamaan kenttäsuunnittelua ja tuomaan maalauksellista jälkeä lopputuotteeseen. Konseptitaidteen maisemia siis käytettiin suoraan pelisuunnittelussa ja graafisessa lopputuloksessa. Ajatuksena oli myös, että visuaalinen ilme olisi niin erottuva, että kaikki tietäisivät *Child of Light*in, jos joku puhuisi "vesiväripelistä". (Plourde 2014) Kuvassa 4 nähdään *Child of Light*in taidetta.



KUVA 4. Kuvakaappaus pelistä *Child of Light*. (The Art of *Child of Light*, GDC Vault 2014). *Child of Light*in konseptitaidetta.

Tässä työssä kuitenkin poiketaan näistä kahdesta ääriesimerkistä ja tutkitaan, kuinka useita erilaisia (niin perinteisiä kuin digitaalisia) tekniikoita voisi käyttää yhtenäisen ja kauniin peliympäristön luomiseen. Perinteisistä tekniikoista valikoitui akryylivärit. Jos perinteisiä tekniikoita aikoo käyttää kaupallisessa peliprojektissa, on kuitenkin syytä tarkastella, onko ratkaisu aikataulun puitteissa järkevä, sillä korjaaminen on joko mahdotonta tai erittäin aikaa vievää. Toisin sanoen maalaus saatetaan joutua aloittamaan alusta virheen sattuessa.

### **2.2.1 Maalausprosessi**

Jotta maalatut ympäristöt olisivat käyttökelpoisia tietokonepelissä, täytyy ennen maalauksen aloittamista ottaa huomioon useita seikkoja. Ehkä tärkeimpänä pelin kuvasuhde. Varsinkin mobiilipeliä suunnitellessa tulee muistaa, että grafiikoiden täytyy toimia useilla eri laitteilla ja useilla eri kuvasuhteilla. Näin ollen ennen kuvan aloittamista, tulee olla tiedossa, minkä kokoinen maalauksesta tulee tehdä. Kuvasuhteen lisäksi kuva kannattaa tehdä myös tarpeeksi suureen kokoon, jotta se siirtyy hyvin myös digitaaliseen muotoon.

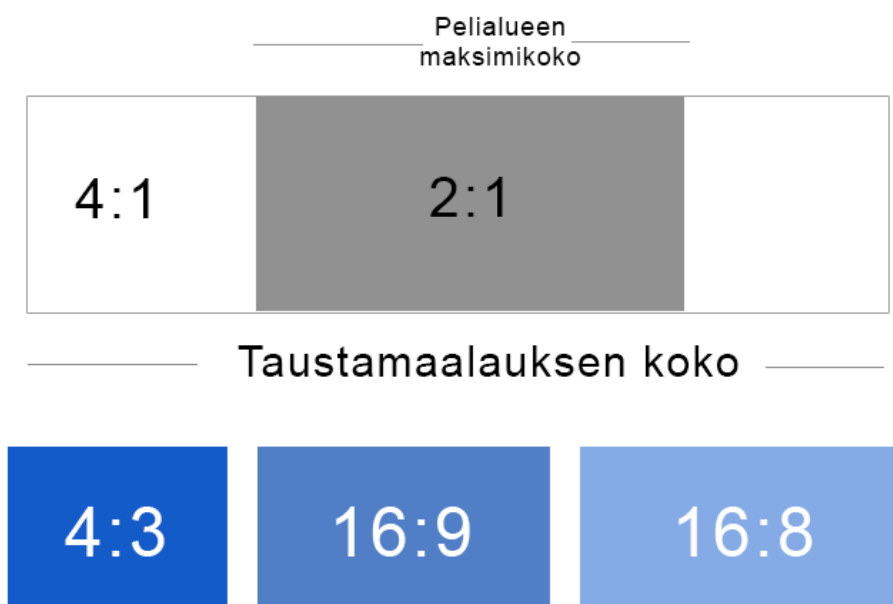
Tärkeää on myös suunnitella tarkkaan pelin graafinen ilme ja tunnelma. Millaisia värejä käytetään ja onko tyyli esimerkiksi sarjakuvamainen, realistinen, maalauksellinen vai jotain muuta. Esimerkkinä Patrick Plourde (2014) haki inspiraationsa *Child of Light*in vanhoista satukirjoista ja niiden kuvituksista. Pelin tuli olla taianomainen ja kuvitusten ja lopullisen grafiikan tuli tukea tätä ajatusta. (Plourde, 2014)

OWLS-pelin ilmeen suunnittelussa inspiraationa toimi Studio Ghiblin 80- ja 90-lukujen animaatiot sekä modernimmat roolipelit ja heavy metal -levyjen kansitaiteet. Värimaailmasta haluttiin rikas, mutta hillitty. Kontrastin kanssa oltiin säästeleitä, jotta pelihahmot ja runsaat erikoistehosteet pääsisivät oikeuksiinsa yksityiskohtaisten ympäristöjen lomassa. Juuri kontrastin vähyyks osassa digitaalisesta grafiikasta loi vaatimuksia käsin tehdyille taustoille. Niiden tuli olla kylläisiä ja syviä. Käsin maalatut pohjat ovat myös pelin ainoat taustakerrokset, jotka ovat

staattisia, joten niiden piti kestää katselua ja toimia sellaisenaan. Romanttisuus ja taianomaisuus olivat myös vaatimuksia maalauksille.

Kuvasuhden tulee aina ottaa huomioon kuvitusta aloittaessa. Täytyy siis olla selvillä siitä, millaisille laitteille peliä ollaan tekemässä. Jackson Wallace mainitsee, että laajakuva ja sen variaatiot ovat yleisimmin käytössä. Laajakuvan kuvasuhde on 16:9 ja tästä voi olla hieman leveämpi malli 16:8. Esimerkiksi vanhojen televisioiden kuvasuhde 4:3 ei ole enää laajalti käytössä uudemmissa laitteissa. (Wallace 2013)

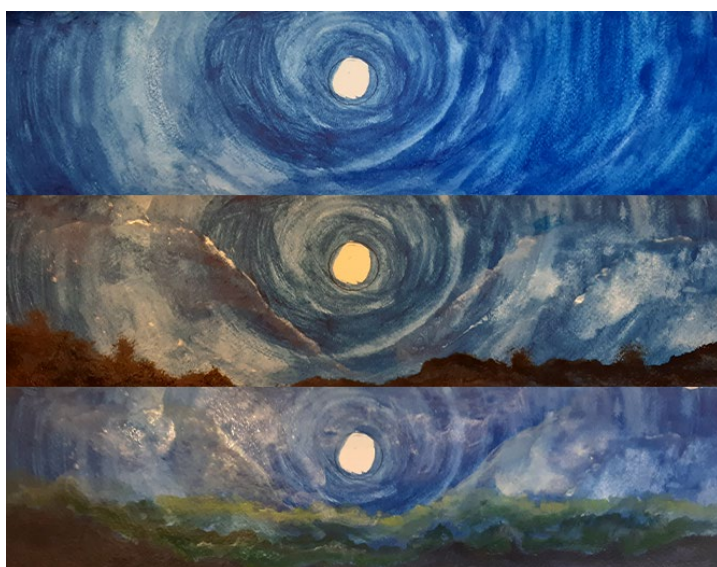
OWLS on mobiilipeli, joten kuvat päädyttiin tekemään kuvasuhteeseen 4:1. Tämä mahdollisti grafiikoiden toimivuuden eri mobiililaitteissa. Tableteille pelialue myös rajattiin siten, että kuvasuhde on sama kuin esimerkiksi puhelimilla, jotta pelikokemus ei kärsi tai muutu. Maalausten valmistuttua huomattiin kuitenkin, että koska maalatut taustat ovat staattisia, ei niistä tulisi näkymään kuin puolet itse pelissä. Tässäkin tapauksessa huolellisempi ennakkosuunnittelu olisi saattanut johtaa vielä parempaan lopputulokseen. Kaaviossa 2 nähdään OWLS-pelin kuvasuhteet sekä yleisimpiä kuvasuhteita.



KAAVIO 1. OWLS-pelin taustojen kuvasuhteet ja pelialueen koko sekä yleisiä kuvasuhteita.

Itse maalaukset toteutettiin akryylimaalauksina akvarellipaperille. Akvarellipaperi alustana luo mielenkiintoista tekstuuria kuviin. Se on havaittavissa yllättävänkin hyvin jopa lopullisessa digitaalisessa versiossa pelissä. Akryylivärit ovat myös siitä kiitollisia, että lopputulosta on mahdollista korjata, joskin hitaasti.

Maalauksissa pyrittiin käyttämään maltillisesti eri värisävyjä. Tämä ei kuitenkaan ollut kovin suuri huolenaihe, sillä joka tapauksessa oli tarkoitus muokata kuvia jälkikäteen digitaalisesti. Näin ollen sävyjä voisi tarvittaessa muuttaa. Aikataulullisesti on kuitenkin järkevämpää tehdä maalaus mahdollisimman lähelle toivottua lopputulosta, jotta maalauksia ei joudu tekemään uudestaan. Digitaalisesta muokkauksesta kerrotaan tarkemmin myöhemmin opinnäytetyössä. Kuvassa 5 nähdään maalausprosessia eri vaiheissa.



KUVA 5. Yhden OWLS-mobiilipelin taustamaalauksen eteneminen.

### 2.2.2 Maalauksesta digitaaliseksi taustaksi

Jotta maalattuja taustakuvia voidaan käyttää pelissä, täytyy ne ensimmäisenä muuttaa digitaaliseen muotoon. Nykypäivänä prosessorit ja näytönohjaimet ovat onneksi niin tehokkaita, ettei maalausjälkeä tarvitse esimerkiksi muuttaa 80- ja 90-lukujen tyyliin pikseligrafiikaksi vaan maalauksen nyanssit saadaan tuotua esiin melko hyvin lopputuloksessa. Kuvien siirtämiseksi digitaaliseen muotoon on kaksi vaihtoehtoa. Kuvat joko skannataan tai kuvataan digitaalisella kameralla.

Skannaus on aina tarkempi ja parempi vaihtoehto, sillä silloin vältetään esimerkiksi valaistusongelmilta, heijastuksilta tai värinästä aiheutuville vääristymille ja epätarkkuuksilta. Skannerilla on myös helppo hallita skannauksen tarkkuutta ja varmistua, että kuvan digitaalinen muoto vastaa käyttötarkoitusta. Jos maalaus on kovin suuri, on mahdollista, että riittävän suurta skanneria ei ole saatavilla ja tällöin kannattaa suosia kameraa. Jos maalauksen joutuu skannaamaan kahdessa osassa, on suuri vaara, että osat eivät yhdisty saumattomasti toisiinsa ilman suurehkoa kuvankäsittelytyötä.

OWLSia varten maalaukset kuvattiin logistisista syistä digitaalisella kameralla ja siirrettiin kamerasta tietokoneelle jatkokäsittelyä varten. Tämä aiheutti jonkin verran jälkityötä muun muassa valoisuuksien korjaamisessa, mutta tärkeintä oli, että kuvat olivat teräviä ja suoria. Kameralla saa kuitenkin pikselikooltaan niin suuria kuvia, että ne ovat enemmän kuin riittäviä peligrafiikkaan.

Kuten aiemmin mainittiin, taustagrafiikoiden kuvasuhteeksi valikoitui 4:1. Näin ollen lopullisen taustagrafiikan kooksi tuli 4800 px x 1200 px. Tämä on riittävä koko kattamaan nykyiset suuresoluutioisetkin mobiilinäytöt.

Työssä kuvankäsittelyohjelmistona käytettiin Adobe Photoshopia. Kun maalaus oli siirretty digitaaliseen muotoon, luotiin Photoshopiin edellä mainitun kokoinen uusi kuvatiedosto ja maalaus tuotiin kuvatiedostoon niin sanottuna Smart Objectina. Smart Object on Photoshopin taso, joka sisältää kuvan kaiken alkuperäisen informaation. Näin ollen kuva on helppo venyttää vastaamaan kuvatiedoston pikselikokoa ja alkuperäiseen kuvaan voi palata, jos tekee muokkauksessa peruuttamattoman virheen. Kun Smart Object oli paikallaan, siitä tehtiin kopio, joka rasteroitiin muokkauksen aloittamiseksi, näin säilyttäen myös alkuperäisen kuvainformaation.

Taustakuvien käsittelyssä on hyvä ymmärtää ainakin Photoshopin väripyörän perusteet. Väripyörän ymmärtäminen auttaa ymmärtämään monia säätötasoja kuten sävyä ja kylläisyyttä. Väripyörä pohjautuu kuuteen perusväriin: syaaniin, siniseen, magentaan, punaiseen, keltaiseen ja vihreään. Väripyörä voidaan jakaa kuuteen osaan näiden värien perusteella. Esimerkiksi mitä lähemmäksi keskiötä

kuvassa 6 näkyvässä pyörässä tullaan, sitä pienempi on kylläisyys. Sävyin valinnassa kuvassa 6 näkyvät viivat kuvaavat astelukuja. Eli punainen on 0 astetta, keltainen 60 astetta, vihreä 120 astetta ja niin edelleen. (Willmore 2004)



KUVA 6. Adobe Photoshopin väripyörä jaettuna karkeasti perusvärien mukaan.

Kuvankäsittelyssä kannattaa aina suosia non-destruktiivista tapaa tehdä muutoksia alkuperäiseen kuvaan. Näin ollen esimerkiksi sävyjen, kylläisyyden, valoisuuden, kirkkauden ja kontrastin muutoksissa kannattaa suosia säätötasoja. Tällöin alkuperäinen kuva ei muutu ja säätötasoja voi muokata tai poistaa mielensä mukaan. Photoshopissa yllä mainitut säädöt on sidottu kahteen eri säätötasoon: sävy/kylläisyys/valoisuus sekä kirkkaus/kontrasti. Nämä olivat myös OWLS-pelin taustojen muokkauksessa pääasialliset säätötasot.

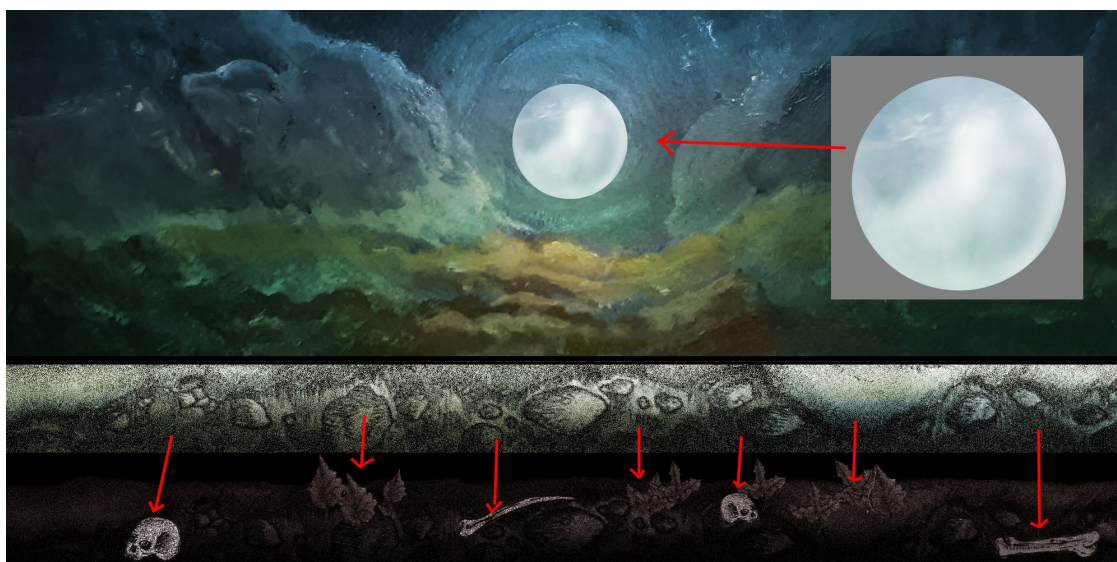
Myös maskien käytön hallinta on oleellinen osa taustakuvien muokkausta. Yksinkertaisuudessaan maskit hyödyntävät tason alpha-kanavaa ja antavat mahdollisuuden päättää, mitkä osat säätötasoista tai kuvatasoista ovat näkyvissä. Näin säätötasojen vaikutuksen voi kohdentaa vain haluamiinsa kohtiin koko kuvan sijaan. Kuvassa 5 violetit alueet on värjätty käyttämällä sävy/kylläisyys/valoisuus-

säätötasoa ja rajoittamalla maskin avulla säätötason vaikutus ainoastaan kuvassa violetteina näkyviin alueisiin. Maskien hyödyntämiseen on monia tekniikoita, mutta lähtökohta on, että maskin toiminta perustuu mustaan, valkoiseen ja harmaansävyihin. Jos maski on kokonaan musta, kuvasta ei näy mitään. Kokonaan valkoinen maski näyttää koko kuvan. Maskin harmaat kohdat näyttävät kuvan osittain läpinäkyvänä. Kuvassa 7 on esimerkki maskin käytöstä ja säätösojen käytöstä.



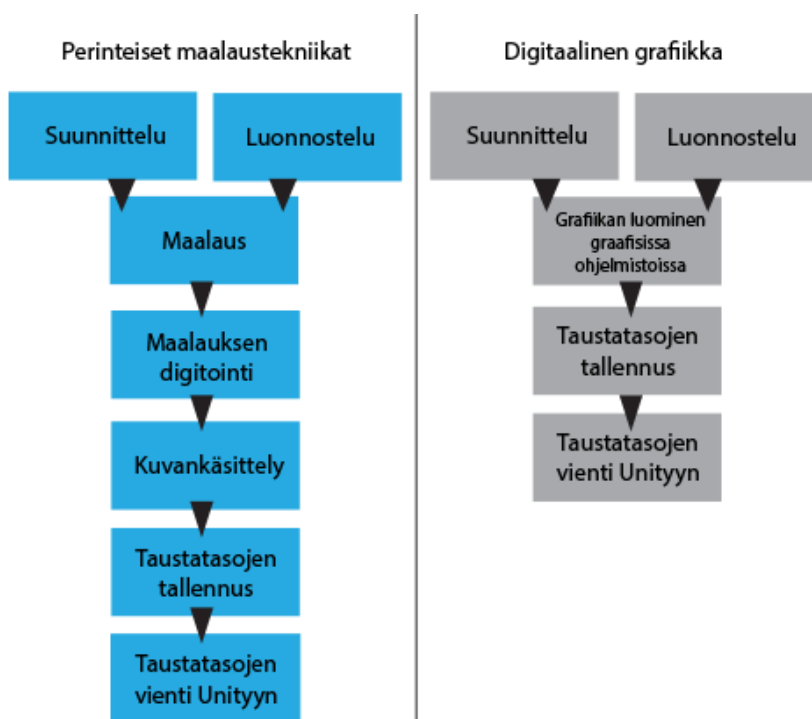
KUVA 7. Säätötasot ja maski.

Maalausten tuomisessa digitaaliseen muotoon on myös se hyvä puoli, että kuviin voidaan jälkepäin tuoda uusia elementtejä ja yksityiskohtia. Perinteisessä taiteessa tämä vertautuu kollaasiin. Tämä mahdollistaa tehokkaamman työnkulun pelitaustoja tehdessä. Kaikkia elementtejä maalaukseen ei ole pakko saada kerralla valmiiksi vaan kuvaan voi halutun tunnelman luomiseksi tuoda uusia näkökulmia yksityiskohtia lisäämällä. Kuvassa 8 on esimerkki, miten maalaus tai piirustus on saatu toimimaan yhtenäisenä kokonaisuutena lisäämällä siihen yksityiskohdiksi pienempiä maalauksia tai piirustuksia.

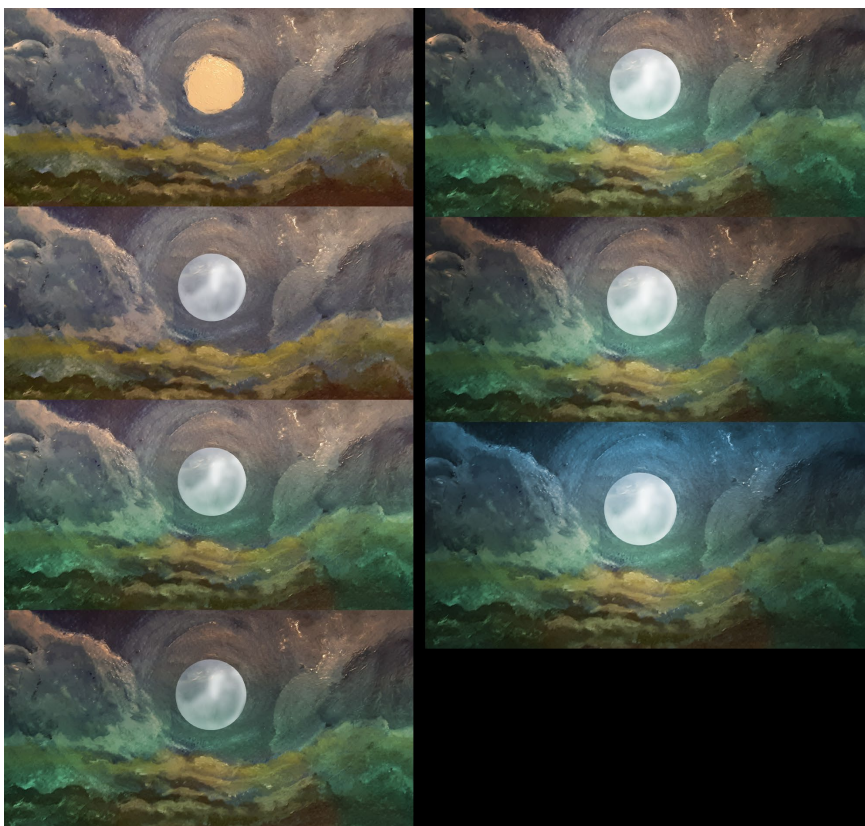


KUVA 8. Taustakuviin on lisätty elementtejä pienemmistä maalauksista.

Edellä mainittujen kuvankäsittelytoimenpiteiden jälkeen lopputulokseksi saatiin tunnelmallinen ja peliin sopiva staattinen tausta, jonka päälle oli hyvä alkaa rakentaa muuta peliympäristöä. Kuva tallennettiin png-formaattiin, jotta sen sai parhaalla tavalla tuotua pelin kehitysympäristöön, Unityyn. Kuvassa 9 on vielä esitetty taustan karkea kehityskaari maalauksesta lopulliseen muotoon. Kaaviossa 2 on vertailtu työnkulkua edellä kuvatussa mallissa sekä pelkästään digitaalisia tekniikoita käyttäen.



KAAVIO 2. Vertailu digitaalisen ja perinteistä maalausta sisältävistä työnkuluista.



KUVA 9. Taustakuvan kehityskaari maalauksesta valmiiksi pelitaustaksi.

### 2.3 Digitaalisesti luodut ympäristöt

OWLS-pelin kaikkia ympäristöjä varten luotiin käsin maalatun staattisen pohjan lisäksi yhteensä viisi vierivää taustatasoa, joiden yhdistelmä loi peliin tunnelmallisen ja tyylikkään peliympäristön. Näiden tasojen luomiseen käytettiin sekä 3D-että 2D-tekniikoita, joita käsitellään seuraavissa luvuissa. Pääasiallisina työkaluina toimivat Adobe Photoshop ja Blender.

Peliin luotiin yhteensä 8 erilaista peliympäristöä. Jokaiseen tarvittiin oma tunnelmansa. Kun projektia aloitettiin, pohdittiin käsin maalaamisen lisäksi parhaita tapoja luoda maisemia ja yksittäisiä puita, aitoja ja muita elementtejä. Myös Unityn 2D-valaistuksen rooli korostui alkuperäisessä suunnittelussa. Hyvin varhaisessa vaiheessa kuitenkin huomattiin, että Unityn 2D-valot vievät yksinkertaisesti liikaa

tehoja näytönohjaimelta mobiilipelissä ja ne eivät yksinkertaisesti tulisi toimimaan. Pitikin miettiä, kuinka saadaan luotua taustoihin upeita valoja ja varjoja ilman varsinaisia valaistusresursseja. Näiden pohdintojen myötä päädyttiin tekniikkaan, jossa Blenderillä luodaan taustoja ja elementtejä pelin taustatasoja varten. Nämä elementit renderoitiin png-kuviksi, jotka sitten siirrettiin Photoshopiin jatkokäsittelyä varten. Tämä mahdollisti valojen ja varjojen hyvän hallinnan Blenderissä ennen renderointivaihetta, mikä toi taustoihin kaivattua syvyyttä. Koska kaikki kahdeksan peliympäristöä sisälsivät vähintäänkin joitain luontoelementtejä, oli myös syytä miettiä, miten niistä saadaan luonnollisen näköisiä. Realismia ei haettu, mutta esimerkiksi muovisen näköiset puut eivät olisi kovin tyylikkäitä.

Kokonaisen luonnon luominen olisi haastavaa, mutta yksittäisten elementtien luominen voi olla kohtuullisen helppoa. Täytyy muistaa, että luodaan peliympäristöä, ei oikeaa luontoa. (Ahearn 2008, 189) Koska OWLS on fantasiateemainen peli, oli ympäristöjen luominen siinäkin mielessä helpompaa. Asioiden mittasuhteet tai realismi eivät olisi oleellisia. Ainoastaan kauneus ja tunnelma merkitsivät.

Suuri haaste ympäristöjen suunnittelussa on tehdä ympäristöstä johdonmukainen, mutta ei itseään toistava. Parhaimmillaan muutaman eri elementin oikealla asettelulla voidaan saada aikaan kokonainen metsä, jossa kaikki puut näyttävät erilaisilta. (Ahearn 2008, 190) Tämä on sekä ajan käytön kannalta tehokasta, että varsinkin 3D-peleissä tehojen optimoinnin kannalta elintärkeää.

### **2.3.1 3D-mallit ja mallien renderointi**

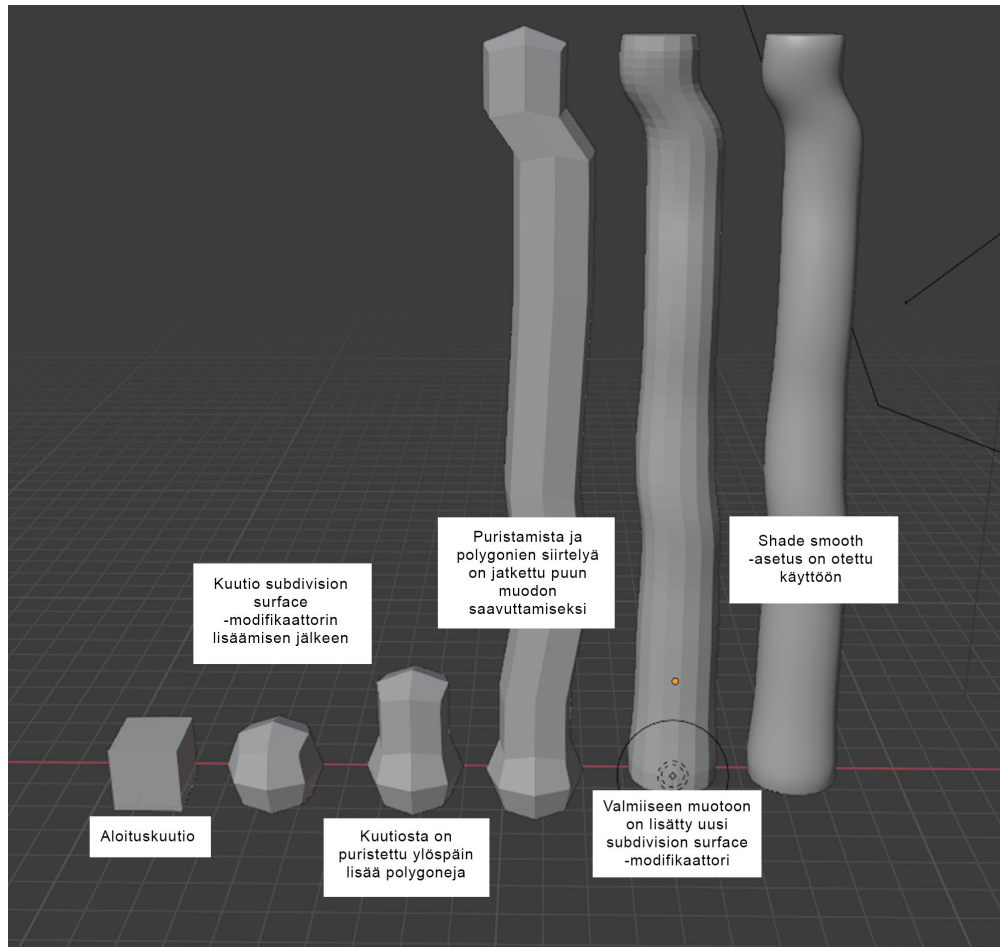
Erilaisten luonnollisten elementtien luominen 3D-ohjelmassa voi olla erittäin aikaa vievä ja hankala prosessi. Puut, kivet ja erilaiset kasvit ovat orgaanisia ja näin ollen niiden piirteiden tulisi olla yksilöllisiä. Luonnollisten elementtien mallintaminen onkin parasta tehdä käsin parhaan lopputuloksen saavuttamiseksi. (Ahearn 2008, 224) Nykyään tarjolla on myös paljon valmiita paketteja, jotka sisältävät 3D-malleja eri tarkoituksiin. Työtä varten kaikki elementit, yhtä Blenderin puutyökalulla tehtyä puuta lukuun ottamatta, luotiin kuitenkin alusta loppuun itse.

Työtä varten oli mallinnettava muun muassa puita, hautakiviä, kiviä, lumikinoksia, pensaita, pylväitä ja raunioita. Jos ympäristön jokainen elementti olisi ollut yksilöllinen ja erilainen, olisi työmäärä ollut valtava. Kuitenkin, koska mallit renderoitiin ja vietiin 2D-maailmaan, oli mahdollista sommitella esimerkiksi muutamaa erilaista puutyyppeä siten, että maisema ei näytä toisteiselta. Koska lopullisena tavoitteena oli myös toistuva parallax scrolling -ympäristö, ei olisi ollut edes mielekäästä luoda valtavan suurta metsää tai muuta maisemaa, koska taustaspritejen leveys oli rajattu aiemmin mainittuun 4800 pikseliin.

Kun mallinnetaan oikeastaan mitä tahansa 3D-objektia 2D-ympäristöä varten, voidaan työprosessi jakaa karkeasti viiteen osaan: mallinnus, UV-karttoitus, teksturointi, valaistus ja renderointi. Kun 3D-mallia ei sellaisenaan käytetä pelissä 3D-ympäristössä, voidaan prosessissa jättää huomiotta asioita, jotka olisivat muuten pelin optimoinnin kannalta tärkeitä. Ei tarvitse miettiä mallin polygonien määrää eikä tarvitse tehdä niin sanottua bakingiä, jotta polygonien määrä saataisiin alhaiseksi, mutta malli riittävän näyttäväksi. Baking olisi siis prosessi, jossa alhaisemman polygonimäärän omaava malli saadaan suuremman polygonimäärän mallin avulla näyttämään yksityiskohtaiselta.

**Mallinnuksessa** aloitetaan yksinkertaisesta muodosta, esimerkiksi kuutiosta, rakentaa haluttua esinettä tai asiaa. Tässä luvussa käytetään esimerkkinä puunrunkoa. Työtä varten piti mallintaa useita puunrunkoja, mutta koska oksien ei tarvinnut näkyä, runko toimii hyvänä yksinkertaisena esimerkkinä niin mallinnuksesta kuin muistakin prosessin vaiheista. Blenderillä mallintaessa hyvä tapa on aloittaa nimenomaan kuutiosta. Varsinkin, jos tekee malleja 3D-ympäristöä varten, kannattaa aloittaa kappaleesta, jossa polygoneja on mahdollisimman pieni määrä. Kuutioon voi sitten lisätä subdivision surface -nimisen modifikaattorin, joka asetuksista riippuen jakaa polygoneja pienempiin osiin. Yksinkertainen subdivision surface kuutiossa toimii hyvänä pohjana puunrunгон mallintamiselle. Subdivision surface täytyy myös muistaa lisätä kuutioon, jotta muokkaamisen voi aloittaa. Tämän jälkeen, muokkaamalla muotoa siirtämällä ja puristamalla polygoneja, voidaan alkuperäinen kuutio muokata haluttuun muotoon. Kuitenkin, koska muokkaus aloitettiin mahdollisimman pienellä polygonimäärällä, on lopuksi vielä syytä käyttää uudestaan subdivision surface -modifikaattoria, jotta puusta

saadaan halutun näköinen, eikä se näytä enää liian kulmikkaalta. Lopuksi täytyy muistaa vielä valita shade smooth -valinta, joka tekee mallista pehmeämmän näköisen. Kuvattu prosessi pääpiirteittäin toimii mihin tahansa kappaleeseen. Toki puunrungon mallintaminen on paljon yksinkertaisempaa kuin vaikkapa kokonaisen rautatiesillan mallintaminen. Kaaviossa 3 näytetään kuution kehitys puunrungoksi.

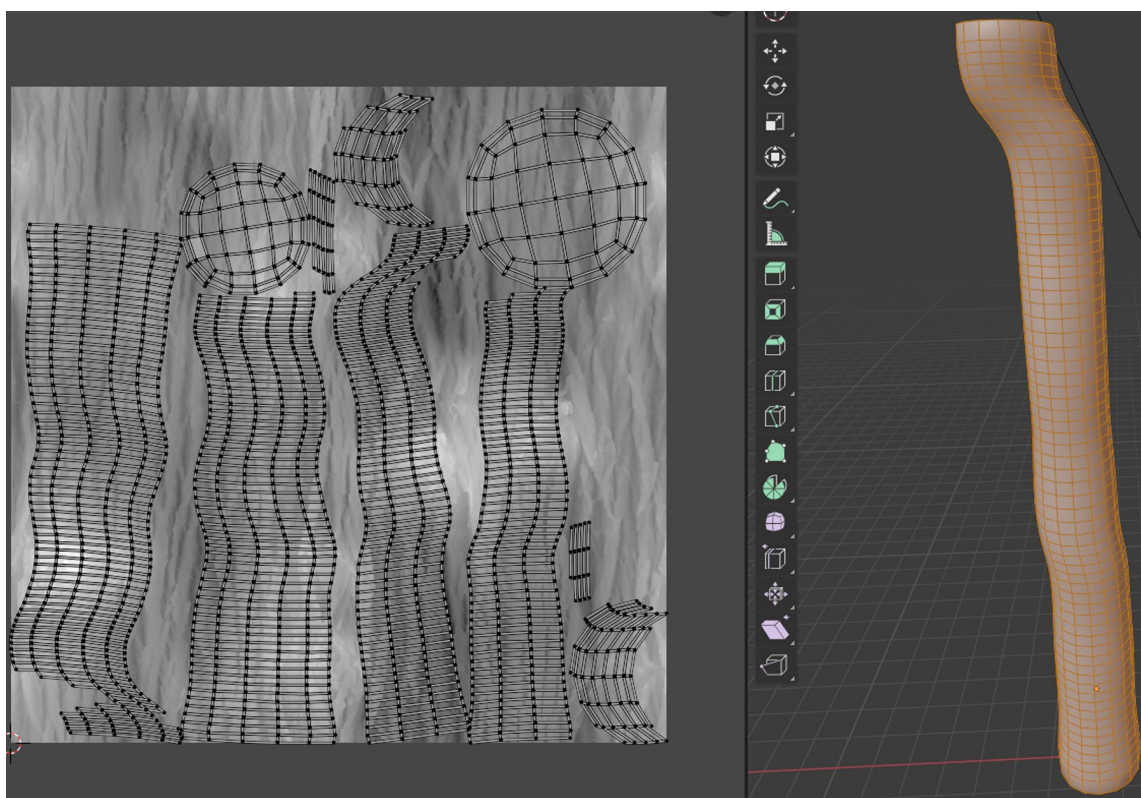


KAAVIO 3. Mallintaminen Blenderissä. Kuutiosta puunrungoksi.

**UV-kartoitus** on prosessi, joka täytyy tehdä ennen varsinaista teksturointia. Ahearn (2008) huomauttaa, että vaikka varsinaisen tekstuurin luominen onkin 2D-prosessi, tekstuurien lisääminen 3D-malliin on oleellinen osa 3D-prosessia. Kartoituksessa määritetään, kuinka 2D-tekstuuri lisätään 3D-mallin pintaan, jotta saadaan vaikutelma jostain tietyistä esineistä tai asiasta. Yksinkertaisimmillaan kuution ympärille kartoitetaan kuva, jotta kuutio näyttäisi joltain tietyltä esineeltä. (Ahearn 2008, 40) Käytännössä siis 3D-malli avataan kaksiulotteiseksi kuvaksi

tai kuviksi ja asetetaan tekstuurina käytettävän kuvan päälle siten, että lopputulos on halutun näköinen.

Blender-ohjelmisto tarjoaa UV-kartoitukseen paljon erilaisia tekniikoita. Koko 3D-mallia ei myöskään tarvitse kartoittaa samalla tekniikalla vaan mallin eri osiin voi soveltaa juuri siihen sopivaa kartoitustekniikkaa. Tässä työssä Blenderin smart unwrap -toiminto ja pieni käsityö tuloksen muokkauksessa osoittautuivat hyviksi toimintatavoiksi. Kuva 10 esittää puunrungon UV-kartan, joka on aikaansaatu edellä mainitulla tekniikalla. Kuvassa nähdään, kuinka puunrungon eri osat on jaettu omiksi kartoikseen, jotta lopputulos näyttäisi mahdollisimmin saumattomalta eikä tekstuuri näyttäisi missään kohtaa puuta venyneeltä.



KUVA 10. Puunrunko ja sen UV-kartta.

**Teksturointi** on 3D-prosessi, jossa luodaan 3D-mallin pintaa esittävä kaksiulotteinen kuva ja liitetään se malliin. Tekstuurit tuovat runsaasti rikkautta ja yksityiskohtia 3D-malleihin (Ahearn 2008, 40). OWLS-pelin ympäristöihin käytettiin polyhaven.com-sivuston valmiita tekstuureita, sillä ne ovat laadukkaita ja täysin vapaasti käytettävissä mihin tahansa tarkoitukseen. Tekstuurit olisi toki voinut luoda itse Photoshopissa, mutta aikataulullisesti valmiiden tekstuureiden käyttäminen

oli järkevämpää. Teksturoinnissa 3D-ympäristöön kannattaa myös kiinnittää huomiota tekstuurikuvien kokoon. Tätä tekstuurien kokoa kutsutaan usein termillä *texel density*. Mitä suurempi on kuvan pikselikoko, sitä enemmän yksityiskohtia voi käyttää, mutta tehonkulutus on myös suurempi. Koska OWLS-pelin mallit siirrettiin 2D-maailmaan, oli tässäkin tapauksessa mahdollisuus käyttää suuria tekstuurikuvia yksityiskohtien saavuttamiseksi. Kuva 11 esittää teksturoidun puunrungon.



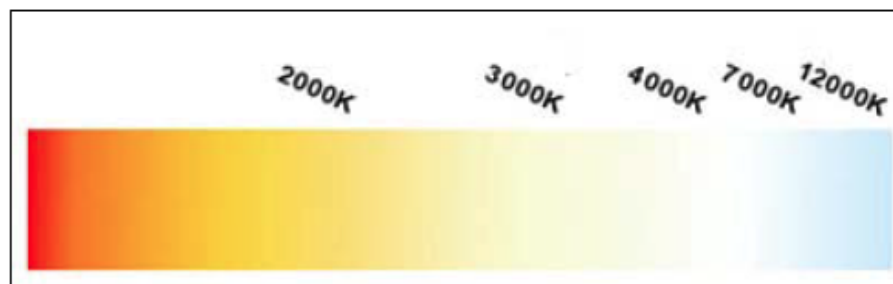
KUVA 11. Puunrunkoon on lisätty tekstuuri, jotta se näyttäisi puulta.

**Valaistus** on mielestäni yksi tärkeimpiä asioita, kun luodaan ympäristöjä peliin. Kuten aiemmin mainittiin, ei OWLSiin ollut mahdollista luoda dynaamista valaistusta tehonkulutuksen vuoksi. Näin ollen 3D-mallien valaiseminen oli ensisijaisen tärkeää oikean tunnelman luomiseksi. Blenderissä on valittavissa erilaisia valovaihtoehtoja: pistevalo, aluevalo, aurinko sekä kohdevalo. Näistä piste- ja kohdevalot olivat käytössä OWLSia luodessa.

Peliympäristöjen valaistusta luodessa on tärkeää miettiä, mistä valo tulee. Mikä ympäristössä on se valonlähde, joka valaistuksen aiheuttaa? Toki esimerkiksi fantasiapelissä valojen realismisuus voi jäädä taianomaisen tunnelman jalkoihin. On kuitenkin hyvä ainakin tiedostaa valonlähteet, jotta ympäristöstä tulisi johdonmukainen. Valo vaikuttaa myös ympäristön syvyysvaikutelmaan, joten senkin takia ainakin pääasiallinen valonlähde on hyvä pitää mielessä valaistusta luodessa.

Jos kuitenkin pyrkii realismiin valaistusta tehdessä, täytyy ensin ymmärtää kuinka värit ja valot korreloivat keskenään oikeassa maailmassa. On tärkeää tietää miten valonlähteet toimivat ja miksi ne toimivat juuri niin. (Powell 2010, 5) Väriteorian hallitseminen on siis oleellista artistille, kun valoja alkaa suunnitella. Ainakin välttävä ymmärrys pääväreistä ja vastaväreistä on tarpeen.

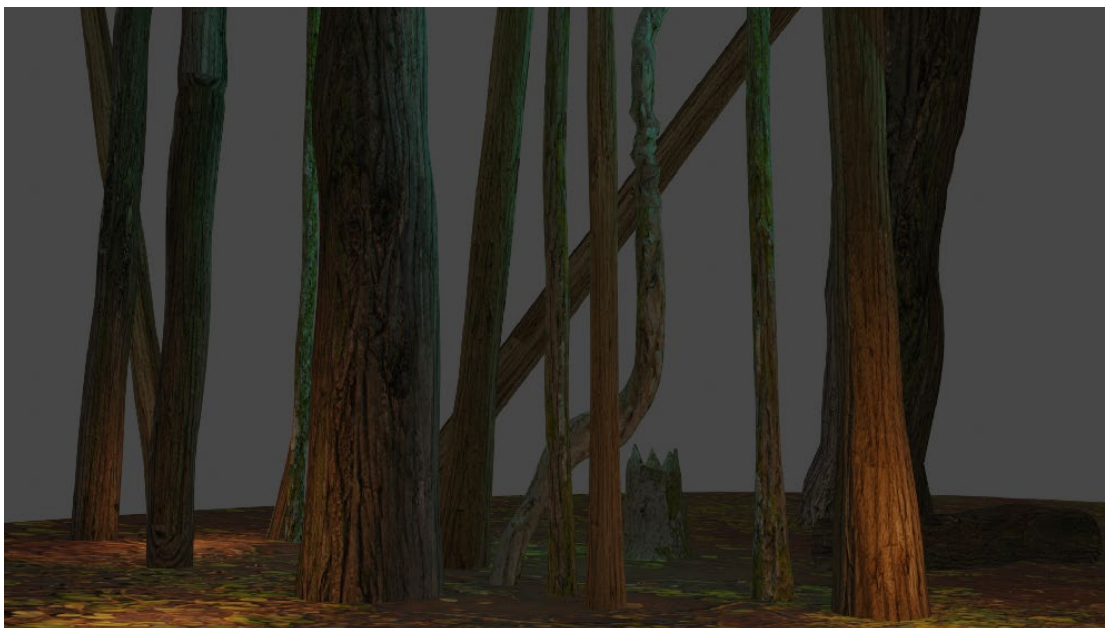
Myös värilämpötilan tunteminen auttaa oikeanlaisen valaistuksen suunnittelussa. Värilämpötila mitataan kelvineissä ja sen väriasteikko vaihtelee sinisestä punaiseen kuvan 12 osoittamalla tavalla. Värilämpötila auttaa ymmärtämään, miksi valonlähteet heijastavat juuri tietynlaista valoa. (Powell 2010, 8)



KUVA 12. Värilämpötila-asteikko. (Blender 2.5 lighting and rendering bring your 3D world to life with lighting, compositing, and rendering. Powell, A.W. 2010). Värilämpötilan asteikon vasemmassa päässä on punainen, oikeassa sininen.

Perinteisessä valaistuksessa käytetään pääasiallisesti kolmea valoa: päävaloa, täytevaloa ja vastavaloa. Päävalo on kuvan pääasiallinen valo ja yleensä valaisee lähes koko kuvan ja määrittää valaistuksen värimaailman. Täytevalo valaisee joitain pimeämpiä kohtia, jotka päävalo aiheuttaa. Takavalon tehtävänä on erottaa kohde taustasta. Se luo kohteen ympärille valoreunuksen. (Powell 21–22)

OWLSia tehdessä pyrittiin pitämään mielessä edellä mainitut konventiot, mutta sadunomaisen tunnelman luomiseksi otettiin vapauksia ja arvioitiin silmämääräisesti, millaiset valot toisivat peliympäristöön halutun tunnelman. Jokaista peliympäristöä tehdessä arvioitiin valaistuksen sävyt ja kirkkaudet sekä värilämpötilat sellaisiksi, että ne parhaalla mahdollisella tavalla tukisivat myös käsin maalattuja taustakuvia ja sopisivat niiden kanssa saumattomasti yhteen. Kuva 134 esittää valaistuja puunrunkoja Blender-ohjelmassa.



KUVA 13. Valaistuja puunrunkoja.

**Renderoinnissa** kolmiulotteisesta datasta tehdään kaksiulotteinen kuva laskutoimitusten avulla (Powell 2010, 30). Blender tarjoaa renderointiin lukuisia asetuksia ja vaihtoehtoja. On mahdollista renderoida kokonainen animaatio, mihin palataan myöhemmin tässä raportissa, tai yksittäisiä kuvia. Peliympäristöjä varten renderoitiin ainoastaan lukuisia valaistuja maisemia ja elementtejä, joista koostettiin lopulliset 2D-taustatasot.

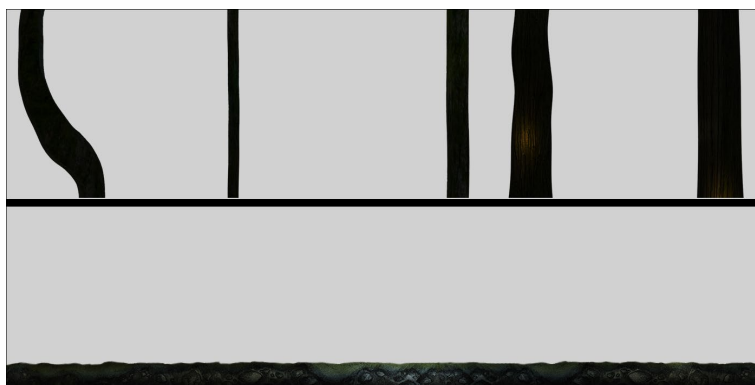
Tärkein asia, joka tulee ennen renderointia valita on renderointimoottori. Renderguide.com-sivusto vertailee artikkelissaan (2022) Blenderin vaihtoehtoja: Eevee ja Cycles. Eevee on reaaliaikainen renderointimoottori ja sitä käytetään usein Blenderissä työskentelyn aikana, jotta voidaan nähdä ainakin suurpiirteinen lopputulos valojen ja tekstuurien kanssa. Eevee ei kuitenkaan anna kovinkaan realistista lopputulosta. Cycles puolestaan on fysiikkaan perustuva moottori ja antaa huomattavasti realistisemmän lopputuloksen. Renderointi Cyclesilla kestää kuitenkin huomattavasti kauemmin. (renderguide.com 2022)

Jos olisi tehty 3D-peli, jossa renderöityjen hahmojen olisi pitänyt toimia suoraan Unityssa, Eevee olisi ollut oikea vaihtoehto. OWLSin taustoja varten olisi voitu valita Cycles. Tämä siksi, että koska tarkoitus oli ainoastaan renderoida kuvia, joista muodostettaisiin 2D-taustakuvia, olisi mahdollista käyttää hitaampaa, mutta realistisempaa jälkeä tuottavaa moottoria. Cyclesin renderointijälki oli kuitenkin

paikoin hieman liian realistista taustoja ajatellen, joten päädyttiin käyttämään sekä Eeveetä että Cyclesiä tilanteen mukaan. Näin lopputuloksesta saatiin halutun näköinen. Jos renderoituja kuvia käytetään ja yhdistellään 2D-kuvankäsittelyohjelmassa, kannattaa ennen renderointia myös valita Blenderin asetuksista filmin laatu läpinäkyväksi. Tämä helpottaa kuvien irrottamista taustasta kompositioita varten. Näiden toimenpiteiden jälkeen taustojen tekemiseen oli kaikki tarvittavat rakennuspalikat. Seuraavassa luvussa käydään läpi prosessi lukuisista renderoiduista kuvista valmiiksi taustoiksi.

### 2.3.2 Renderoinneista taustaspriteiksi

Lopulliset kuvatasot peliympäristöä varten luodaan Photoshopissa tai vastaavassa 2D-kuvankäsittelyohjelmassa. Tässä luvussa käydään läpi peliympäristön valmiiksi saattaminen OWLS-pelin metsäympäristön esimerkin avulla. Lopullisena tavoitteena on siis aiemmin käsitellyn käsin maalatun staattisen taustan lisäksi luoda 5 kuvatasoa, joiden ohjelmallinen liikuttelu pelissä on mahdollista. Haasteena tällaisessa kompositiossa on, että taustakuvista pitää saada saumattomasti itseään toistavia. Eli kuvan oikean ja vasemman reunan pitää sopia yhteen, jotta niitä voi toistaa yhä uudestaan ilman, että pelaaja kiinnittää saumoihin huomiota. Tämä on helppoa niissä tasoissa, joissa kuva-alue leveysuunnassa ei ole täyteen ahdettu, esimerkiksi tasossa, jossa on vain puunrunkoja. Tällöin oikea ja vasen reuna voidaan jättää tyhjäksi saumojen välttämiseksi. Toisin asia on esimerkiksi maatasossa, jossa maan pitää jatkua ruudun vasemmasta reunasta aina oikeaan asti. Kuvassa 14 nähdään ero näiden kahden välillä.



KUVA 14. Kaksi esimerkkiä taustakuvatasoista. Ylhäällä oikea ja vasen reuna ovat tyhjiä. Alhaalla kuva jatkuu reunasta reunaan.

Kun lähdetään toteuttamaan taustatasoja Photoshopissa, on suositeltavaa tehdä ne kaikki kansioituina samaan psd-tiedostoon. Jokaisella viidestä tasosta on siis oma kansio psd-tiedostossa. Näin siksi, että tällöin on mahdollista nähdä kaikki tasot yhdessä tai erikseen sekä tehdä lopuksi säätötasoja, jotka vaikuttavat kaikkiin kuvatasoihin. Näin varmistutaan siitä, että kaikki taustatasot sopivat yhteen. Jos kuvat renderoidaan 3D-ohjelmassa juuri sellaiseksi kuin ne taustaan halutaan, on mahdollista tuoda ne suoraan psd-tiedostoon Smart Objecteina. On kuitenkin todennäköisempää, ehkä jopa suotavampaa, että renderoidut kuvat avataan Photoshopissa ja niistä irrotetaan elementtejä, joita sitten tuodaan psd-tiedostoon, jossa lopullisia kompositioita tehdään. Näin voidaan siirrellä yksittäisiä elementtejä, kuten puita, haluamilleen paikoille ja on helpompi hallita lopullista sommittelua. Kun elementit ovat erillisiä, on myös mahdollisuus monistaa halutessaan joitain elementtejä ja käyttää niitä useammassa kohdassa kompositiota.

Tasojen saumattomuuden varmistamiseen on varmasti monia eri tekniikoita. Työssä käytettiin melko takuuvarmaa tapaa saada tasot saumattomiksi. Kun kompositio on halutun näköinen, toisesta reunasta kopioidaan kaikki sisältö pieneltä leveydeltä. Tämän jälkeen kopioitu alue liitetään osaksi kuvaa omana tasonaan. Taso käännetään vaakasuunnassa peilikuvaksi, jolloin saadaan ikään kuin vastakkainen reuna aikaiseksi. Taso ei tietenkään sellaisenaan sovi toiseen reunaan. Photoshopin maskit ja patch-työkalu ovat erinomaisia apuvälineitä, jotta saadaan reuna sopimaan saumattomasti muuhun taustaan. Maskien avulla liitetyn tason kuvaan sopimatonta reunaa voi häivyttää ja patch-työkalulla voi hakea liitettyyn palaan sellaista tekstuuria, että se ei näytä aivan samalta kuin toinen reuna. Aivan äärilaita täytyy tietysti jättää koskemattomaksi, jotta haluttu saumattomuus säilyy. Näin saadaan vaikutelma pelissä, että tausta jatkuu loputtomasti, vaikka tosiasiallisesti toistetaan vain samaa kuvaa yhä uudestaan.

Lopuksi täytyy vielä katsoa kaikkia tasoja päällekkäin ja tehdä haluttuja sävy-  
muokkauksia säätötasoja käyttäen. Kun lopputulos on haluttu, voidaan tasokansioita katsoa yksi kerrallaan ja tallentaa peliä varten omiksi kuvikseen eli taustaspriteiksi. On erityisen tärkeää käyttää tallennuksessa png-formaattia, sillä taustojen tulee olla läpinäkyviä niiltä osin, missä kuvainformaatiota ei ole. JPEG-for-

maatti ei tähän pysty. Kuvassa 15 on esitetty valmiit kuvatasot erikseen ja kuvassa 16 nähdään, kuinka ne sopivat päällekkäin ja luovat valmiin peliympäristön. Huomion arvoista on vielä, että päällimmäinen taso tulee pelissä kaikkein lähimmäksi pelaajaa. Pelihahmot toimivat sen takana, jolloin kolmiulotteinen vaikutelma korostuu entisestään. Seuraavassa luvussa puhutaan lisää taustojen syvyystvaikutelmasta ja liikkeestä.



KUVA 15. Valmiit kuvatasot erillisinä kuvina.

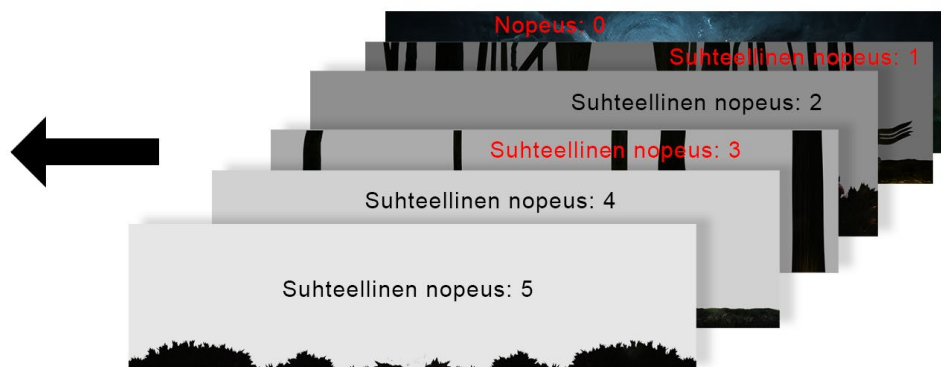


KUVA 16. Valmis peliympäristö.

### 2.3.3 Parallax scrolling

Takashi Nishiyaman Moon Patrolia pidetään yleisesti ensimmäisenä pelinä, joka käytti parallax scrolling -tekniikkaa. Tekniikkaa on kuitenkin käytetty animaatio-elokuvissa jo 1930-luvulta lähtien. Monitasokameraa käyttäen oli mahdollista saada aikaan kolmiulotteinen vaikutelma, jossa animaatioon luotiin syvyyttä liikuttamalla kuvia eri nopeuksilla suhteessa hahmotettuun linssin etäisyyteen. Samaa tekniikkaa hyödynnetään myös tietokonepeleissä, mutta monitasokameran sijaan kuvat asetetaan pelissä eri kerroksiin ja tarvitaan vain yksi kamera. (Bone 2014)

Edellisessä luvussa käytiin läpi eri kuvatasojen luominen. Nämä kuvatasot soveltuvat erinomaisesti parallax scrolling -taustojen toteuttamiseen. Koska tasoja voidaan toistaa yhä uudestaan, niitä voidaan ohjelmallisesti liikuttaa oikealta vasemmalle eri nopeuksilla, jolloin saadaan vaikutelma sekä syvyydestä että paikallaan olevan pelihahmon liikkeestä. Jotta syvyysvaikutelma saavutetaan, etualalla olevien tasojen nopeuden täytyy olla suurempi kuin taka-alalla olevien. Kaavio 4 havainnollistaa tasojen asettelua ja nopeutta syvyysvaikutelman ja liikkeen luomiseksi.



KAAVIO 4. Kuvatasojen asettelu ja nopeus parallax scrolling -ympäristössä.

### 3 PELIHAHMOJEN LUOMINEN JA ANIMOINTI

#### 3.1 Pelihahmojen luominen

Lukuisia pelihahmoja on esiintynyt eri tyyllisissä ja eri genrejen peleissä vuosien varrella ja osa niistä on saavuttanut jopa suurta suosiota eri puolilla maailmaa. Varhainen hahmo sattoi olla esimerkiksi pelkkä piste, joka oli esittävinään hiirtä vuonna 1959 MIT:ssä kehitetyssä pelissä *Mouse in a Maze*. Ensimmäisiä tunnettuja hahmoja, joilla jo oli visuaalinen identiteetti, oli vuonna 1980 esitelty *Pacman* ja sitä jahtaavat kummitukset. Monet pelituottajat, jotka ovat pystyneet luomaan hahmolle riittävän vahvan visuaalisen identiteetin, ovat myös pystyneet kaupallistamaan pelihahmon ja myymään erilaisia oheistuotteita hahmon suosion vuoksi. (Sloan 2015)

Pelihahmot ovat tärkeä osa pelin visuaalista identiteettiä. Monesti pelien viehätys kuluttajalle perustuu nimenomaan pelihahmoihin. Moni tunnistaa esimerkiksi 80-luvun alussa luodun *Super Marion*. Suomalaisen Rovion luomat vihaiset lintuhahmot *Angry Birds* -pelissä ovat myös nousseet maailmanlaajuiseen suosioon.

Pelihahmojen luominen voi olla monimutkainen prosessi. Se ei koostu pelkästään kiinnostavasta grafiikasta vaan myös hyvästä kirjoitustyöstä. Pelihahmon ulkonäkö ja jopa animaatiot voivat saada vaikutteita pelin taustatarinasta. Hyvän kirjoittamisen jälkeen on mahdollista tehdä hahmoista konseptipiirustuksia, jotka määrittävät hahmon ulkonäköä ja toimivat referensseinä esimerkiksi 3D-mallinnuksessa. (Totten 2012, 4)

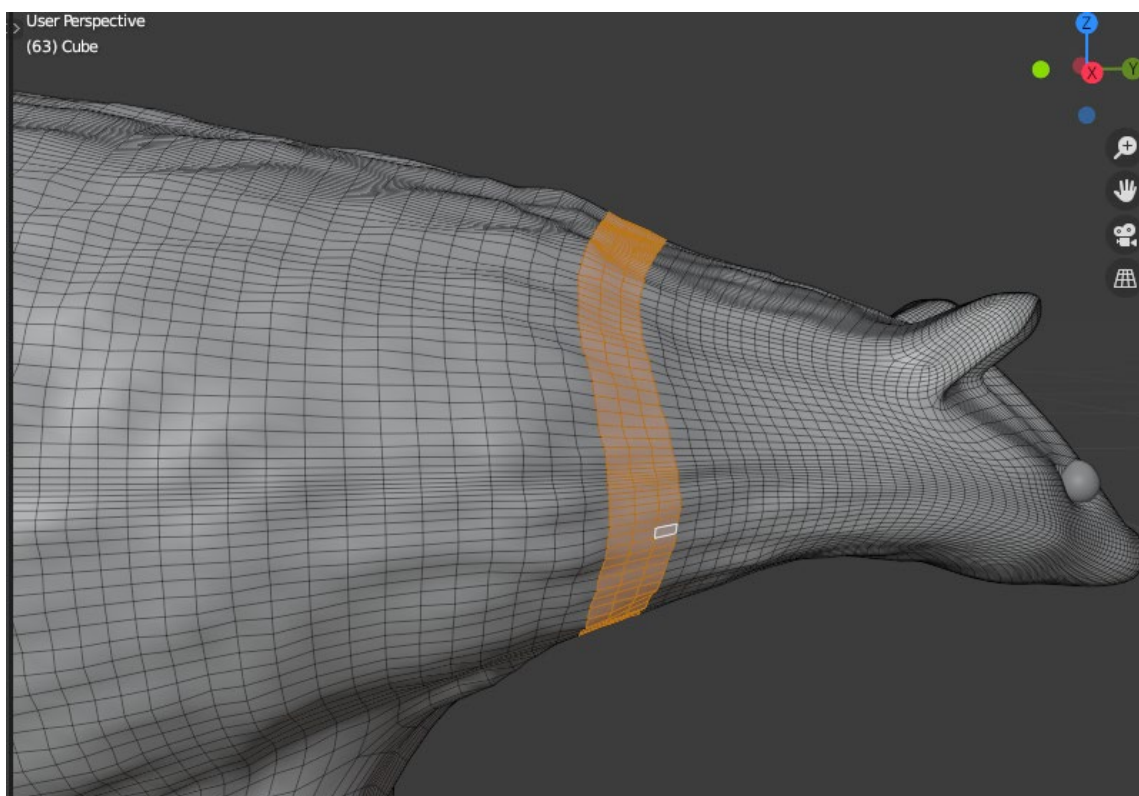
Tarinan ja hahmojen taustatarinoiden kirjoittaminen on melko suuri prosessi. On myös pelin tyylistä ja laajuudesta kiinni, kuinka paljon kirjoitettua materiaalia hahmoja varten tarvitaan. OWLS-pelissä lähinnä aikataulun ja mobiilipelin tyylin vuoksi hahmojen luominen aloitettiin lähes suoraan 3D-mallinnuksesta. Tarina ja hahmot syntyivät tavallaan käsi kädessä. Kun oli luotu pelin päähahmo mäyrä ja muutamia pöllövihollisia, tarina alkoi muodostua näiden hahmojen ympärille. Tämä on ehkä hieman takaperoinen tapa luoda hahmoja ja pelin tunnelmaa, mutta lopputulos on yllättävän toimiva.

Hahmon suunnittelussa ja erityisesti toteutuksessa on tärkeää, että hahmo erotuu taustastaan ja pääsee oikeuksiinsa sille asetetussa peliympäristössä. Vaikka visuaalisen tyylin tulee olla johdonmukainen, on monesti hyvä erottaa hahmojen ja peliympäristöjen tyyli hieman toisistaan, jolloin hahmo saa selkeän ja näkyvän paikan pelissä. Jo pelattavuudenkin kannalta on tärkeää, ettei hahmo huku taustansa. Esimerkiksi aiemmissa luvuissa läpikäyty OWLS-pelin melko maalaukselliset taustat veivät hahmosuunnittelun hyvin selkeään ja hieman yksinkertaisempaan suuntaan. Hahmoissa ei käytetty juuri lainkaan tekstuureita vaan yksinkertaiset ja tyylikkää materiaalit 3D-malleissa tuottivat halutun lopputuloksen. Yksinkertaisuus ei tässä tapauksessa tarkoita siis sitä, etteikö hahmojen malleissa olisi ollut yksityiskohtia. Mallien ulkoasu saatiin erottuvaksi nimenomaan värityksen tyyllillä, sillä se poikkesi vahvasti taustoista sopien kuitenkin värimaailmaltaan pelin ulkoasuun.

Kuten taustaelementeissäkkin, pelihahmojen 3D-mallinnus on hyvä aloittaa kuutiosta. Kuutio tarjoaa hyvän topologian hallinnan ja polygonien määrä ei pääse kasvamaan liian suureksi (Totten 2012, 48). Tämä on erityisen tärkeää silloin, kun 3D-malleja käytetään pelissä sellaisenaan. 2D-pelissä työnkulku on toisenlainen eikä polygonien määrällä ole juurikaan merkitystä. Topologia on tässäkin tapauksessa kuitenkin asia, joka tulee ottaa huomioon, sillä 3D-animaatiot vaativat hahmolta vähintäänkin välttävää topologiaa onnistuakseen.

Hahmoa luodessa voi olla hyvä käyttää myös referenssikuvia, esimerkiksi konseptikuvia, joita peliä varten on luotu. OWLSin tapauksessa internetistä napatut eläinkuvat toimivat hyvinä referensseinä. Referenssikuvissa on tärkeää, että hahmo on kuvattu sivulta, takaa ja edestä, jolloin mallista saadaan joka puolelta oikean näköinen. Esimerkiksi fantasiateema tai se että ei pyritä realismiin antaa kuitenkin mallintajalle vapauksia hahmon suhteen. Kuutiosta on helppo referenssikuvien avulla lähteä muotoilemaan hahmon torsoa. Puristamalla ja jakamalla polygoneja pienempiin osiin voi koko ajan hallita polygonien määrää ja topologiaa ja alkaa muodostaa referenssikuvan mukaista muotoa. Aluksi muoto voi olla karkeakin, mutta mielestäni kannattaa tehdä pohjatyö muistuttamaan referenssiä niin hyvin kuin mahdollista.

Kun torso on saatu muodoltaan referenssin mukaiseksi tai muuten halutun näköiseksi, on seuraavaksi muotoiltava hahmon pää ja raajat. Pään muotoilu onnistuu useimmiten helposti samalla tavalla kuin torsoa on muotoiltu. Pitää kuitenkin kiinnittää huomiota siihen, että torson ja pään välissä kaulan topologia on siisti. Tämä siksi, että sekava polygonirakenne kohdissa, joista hahmoa pitäisi olla mahdollisuus animoida, voi aiheuttaa hankaluuksia animointivaiheessa. Yksinkertainen rakenne, mutta riittävä määrä polygonikerroksia näissä taitekohdissa on tie sujuvaan animointiin. Kuvassa 17 on korostettuna esimerkki siististä taitekohdan topologiasta.

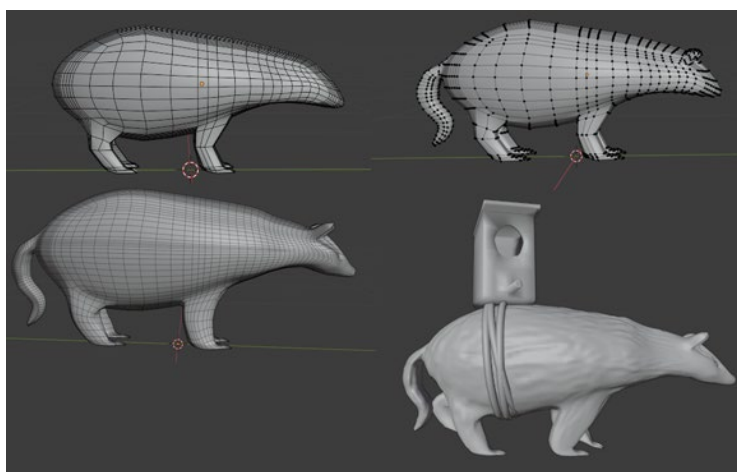


KUVA 17. Mäyrähahmon kaulan topologiaa.

Raajojen luomiseen paras keino on puristaa ne ulos torsosta. Saattaisi olla houkutteleva ajatus tehdä raajat erikseen, mutta raajojen tulisi olla mieluiten kiinni torsossa animaatioita varten. On toki mahdollista myös mallintaa raajat erikseen ja esimerkiksi Blenderin Bridge-toiminnolla kiinnittää ne torsioon. Näin toimittiin esimerkiksi OWLSin pöllöjen siipiä luodessa. Tällainen työtapa kuitenkin useimmiten aiheuttaa lisätyötä. Sen sijaan raajojen puristaminen torsosta käyttäen

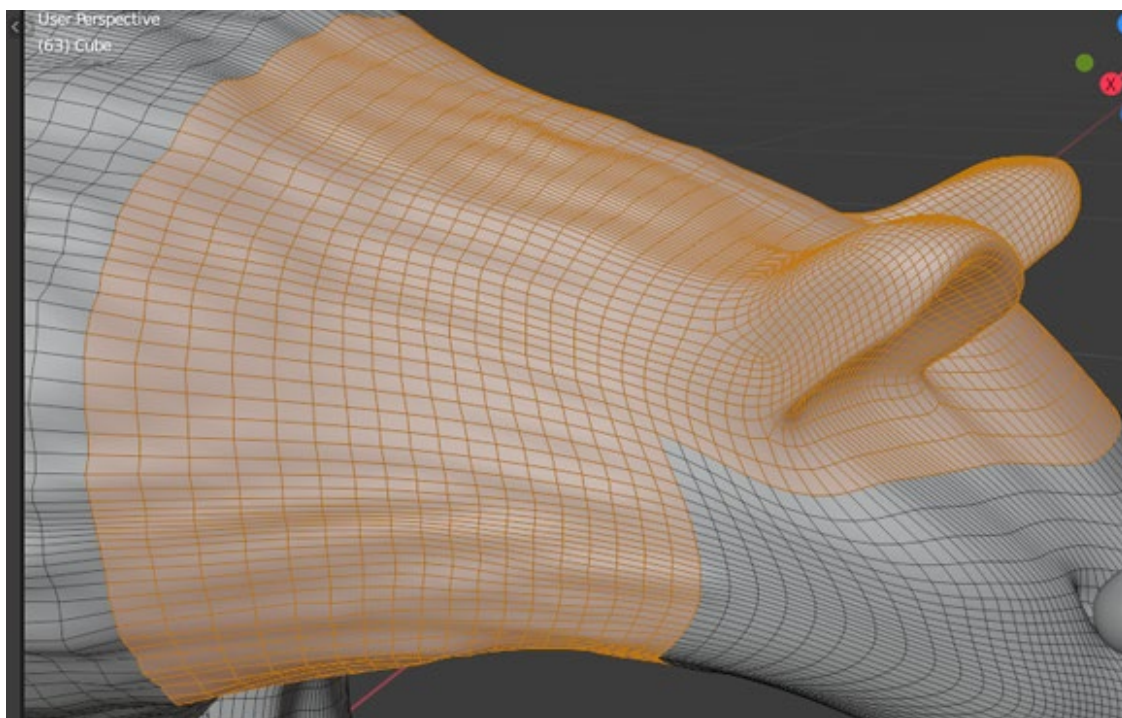
Blenderin symmetriatoimintoja mahdollistaa viereisten raajojen tekemisen samalla kertaa, jolloin niistä tulee automaattisesti identtisiä. Myös hahmon mahdollinen häntä, sarvet tai muut tällaiset ruumiinosat on hyvä tehdä raajojen tapaan puristamalla ne torsosta. Jos kuitenkin haluaa hahmoon lisää realismia ja luonnetta, kannattaa esimerkiksi silmät ja silmäluomet luoda erillisinä objekteina. Myös hahmon mahdolliset lisävarusteet, kuten aseet, kannattaa mallintaa erikseen. Esimerkiksi kuvassa 16 näkyvä mäyrän linnunpönttö ja köydet ovat erillisiä objekteja.

Kun hahmon perusmuoto on kaikin puolin valmis, sitä voi alkaa muokata huolitellummaksi. Jo aiemmin työssä mainittu Subdivision surface -toiminto voi olla monessa tapauksessa hyvä tapa tuoda hahmoon pehmeyttä ja se myös mahdollistaa paremman yksityiskohtien muokkauksen, sillä se lisää polygonien määrää. Tämä on erityisesti hyvä tapauksessa, jossa mallit ja animaatiot viedään lopulta kaksiuolotteiseen maailmaan. Hahmon muokkausta voidaan jatkaa Blenderin Sculpt-toiminnoilla, jotka nimensä mukaisesti toimivat muovailun ja kuvanveiston tavoin. Juuri muovauksen aikana suuresta polygonien määrästä on hyötyä, sillä se mahdollistaa hahmon yksityiskohtaisemman muovauksen. Muovauksen aikana hahmoon voi tuoda erilaisia yksityiskohtia intuitiivisesti. Tässä vaiheessa on hyvä myös muistaa, että esimerkiksi luonnossa harvat asiat ovat täysin symmetrisiä. Näin ollen juuri muovauksen aikana kannattaa tehdä hahmosta epäsymmetrinen pienillä yksityiskohdilla. Esimerkiksi OWLSissa mäyrän turkin laineet eivät ole samannäköinen molemmissa kyljissä. Kuvassa 18 havainnollistetaan pelihahmon mallintamisen eri vaiheita.



KUVA 18. Mäyrähahmo mallintamisen eri vaiheissa.

Tärkeä osa hahmon visuaalista identiteettiä on väryty. 3D-malleissa käytetään joko värillisiä materiaaleja tai niihin voidaan lisätä yksityiskohtaisempia tekstuuria, kuten aiemmin työssä käytiin läpi. Huolellinen topologia auttaa myös hahmon värittämisessä. Kun topologia ei ole liian monimutkainen, eri alueiden valitseminen hahmosta helpottuu. Jos esimerkiksi halutaan tehdä hahmon ihonväri tietynlaiseksi, voidaan valita kaikki ihonväriset alueet ja tehdä niistä Blenderissä ns. Vertex group. Tällöin valinta tallentuu ja siihen voi myöhemmin palata helposti, jos haluaa esimerkiksi vaihtaa ihon materiaalin. Vertex groupiin voi myös koska tahansa lisätä polygoneja tai siitä voi poistaa halutun osan. Kuvassa 19 nähdään esimerkki Vertex groupista.



KUVA 19. Esimerkki Vertex groupista.

Kun malli on jaettu haluttuihin Vertex groupeihin, voidaan jokaiselle ryhmälle antaa oma materiaali. Näitä materiaaleja muokkaamalla on helppo nähdä, miten hahmon värit ja materiaalit toimivat keskenään. Värien lisäksi materiaaleista on helppo muokata muitakin ominaisuuksia, kuten heijastuksia, kiiltoa, metallisuutta ja niin edespäin. Materiaaleihin voi myös halutessaan lisätä kuvatekstuurin. Tämä työnkulku hahmon materiaalien lisäämisessä on artistille hyvä juuri siksi, että työn jäljen näkee välittömästi ja tyyliä ja ilmettä on helppo muokata mie-

leiseksi. Kun materiaalit ovat valmiit, hahmo täytyy vielä valaista. Hahmon valaisemiseen pätevät samat lainalaisuudet, joita kuvattiin ympäristön valaisemisen yhteydessä aiemmin tässä opinnäytetyössä. Kuvassa 20 esitellään Mäyrä-hahmo valmiissa, valaistussa muodossaan.



KUVA 20. Mäyrä valaistuna juhlapuvussaan.

Staattinen hahmo ei kuitenkaan sellaisenaan riitä peliin. Jotta hahmolle saadaan luonnetta ja se saadaan osaksi peliä, se tulee animoida. Liike tuo hahmoon elämää ja siitä tulee uskottava osa pelimaailmaa. Olisi hieman turhaa luoda elävä ja tunnelmallinen peliympäristö, jos hahmot olisivat staattisia ja muistuttaisivat enemmänkin pahvimalleja. Seuraavassa luvussa käydään läpi 3D-mallien animoinnin työkulkua ja tarkastellaan, kuinka animaatiot tuodaan 2D-maailmaan eläväksi osaksi pelin ulkoasua.

## 3.2 Pelihahmojen animaatiot

Animointi on uskottavan pelihahmon luomisen viimeinen ja kaikkein monimutkaisin osa. Jotta 3D-hahmo saadaan liikkumaan, se täytyy rigata, eli hahmolle luodaan animaatioluuranko, jonka avulla se liikkuu. Animaatioluuranko toimii paljolti samoin tavoin kuin savihahmon sisälle tehty rautalankaluuranko. Katsoja näkee vain päällä olevan saven, mutta animaattori liikuttelee rautalankaa hahmon sisällä. (Totten 2012, 157–158)

Pelihahmojen animoinnissa korostuvat yksittäiset liikesarjat kokonaisien animaatiokohtauksien sijaan. Yleisiä liikkeitä ovat esimerkiksi kävely (walk cycle), erilaiset hyppyt ja lyönnit sekä animaatio, joka näytetään, kun hahmo on joutilas, eli niin sanottu idle animation. Näitä liikesarjoja kutsutaan toiminnoiksi, actioneiksi. Idle animation on tärkeä osa hahmon animointia, sillä tämän avulla hahmo näyttää eläväiseltä, vaikka se ei juuri sillä hetkellä tekisikään yhtään mitään. Näitä toimintoja voidaan pelimoottorissa kontrolloida ohjelmallisesti ja niitä voidaan liittää ohjaimen tai näppäimistön komentoihin, jolloin pelaaja voi ohjata hahmoa.

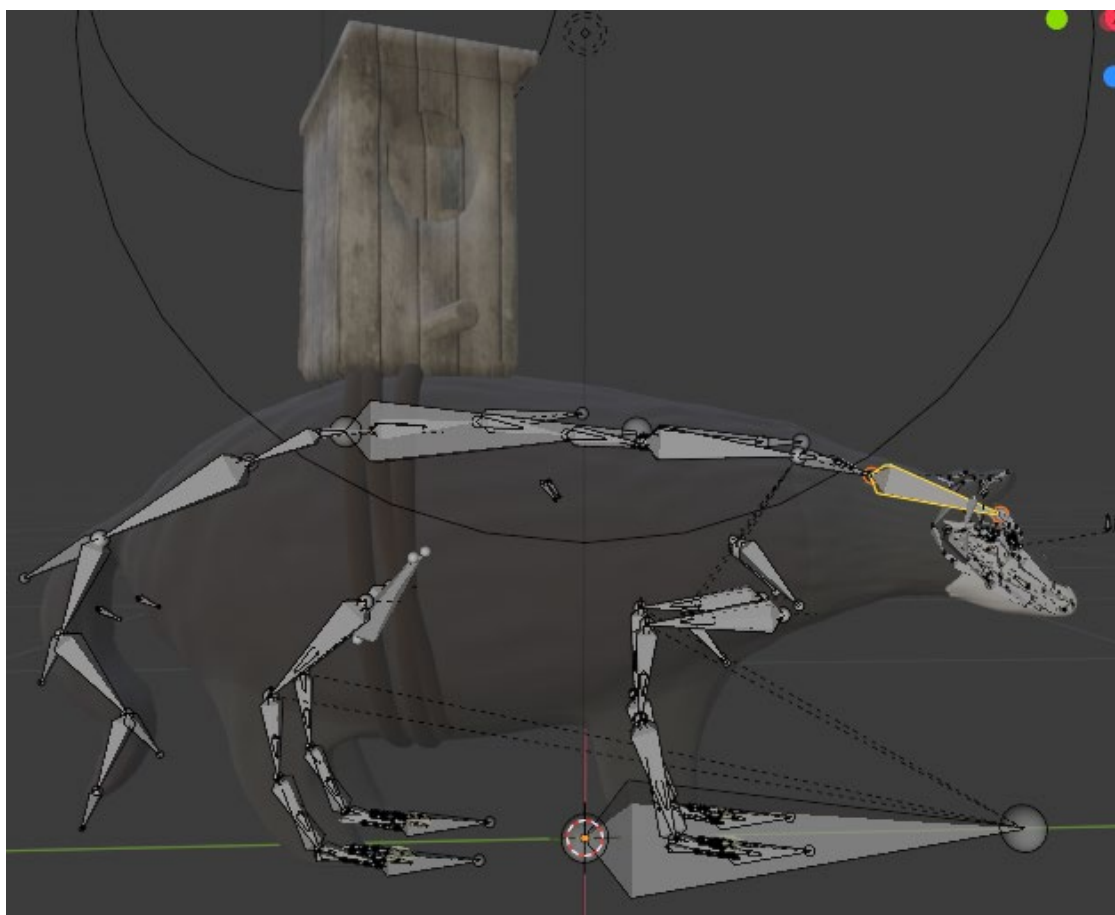
Kun 3D-animaatioita käytetään 2D-pelissä, täytyy ne animoinnin jälkeen renderoida kuvasarjaksi, josta valitaan sopiva määrä kuvia sprite-animaation muodostamiseksi. Seuraavissa luvuissa käydään läpi vaiheet, joiden avulla näytävät animaatiot saadaan luotua 2D-peliä varten.

### 3.2.1 Riggaus

Riggauksella tarkoitetaan kontrolliobjektien luomista, joilla muokataan toista objektia, tässä tapauksessa pelihahmoa. Nämä kontrolliobjektit muodostavat jo edellä mainitun animaatioluurangon. Riggausta pidetään yleisesti ottaen yhtenä vaikeimmista vaiheista 3D-animaation tekemisessä. Riggausta tehdessä on hyvä muistaa kaksi pääasiaa. Animaatioluuranko ei saa olla liian monimutkainen, jotta animaattori pystyy sitä järkevästi liikuttelemaan. Se ei myöskään saa olla liian

yksinkertainen, jotta uskottavat liikkeet on mahdollista toteuttaa. Koska jokaisessa animaatioprojektissa on omat haasteensa, riggaukseen ei ole yhtä oikeaa tapaa. (Vasconcelos 2011, xxiv)

Nykyään on saatavilla erilaisia apuvälineitä riggauksen helpottamiseksi. OWLS-pelin animaatioissa käytettiin Blenderin Rigify-lisäosaa, jolla on mahdollista luoda animaatioluurankoja tai luuketjuja automatisoidusti. Rigify sisältää luurankoja muun muassa ihmiselle, nelijalkaiselle olenolle ja linnulle. On kuitenkin hyvä ymmärtää riggauksen perusteet, jotta animointi onnistuu sujuvasti, käytti sitten lisäosia tai ei. Kuvassa 21 on havainnollistettu Mäyrän animaatioluuranko.



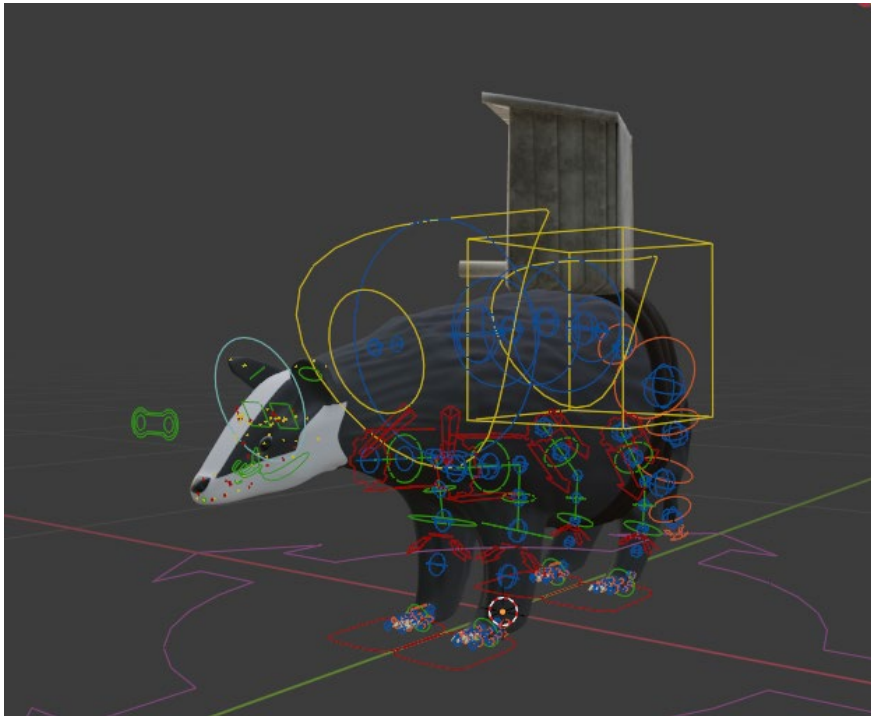
KUVA 21. Mäyrän Rigify-luuranko.

Blenderissä voi luoda yksittäisiä luita, joilla kaikilla on kolme osaa: pää, runko ja häntä. Luita voi puristaa lisää olemassa olevan luun päästä tai hännästä. Näin luodusta uudesta luusta tulee alkuperäisen lapsi, mikä tarkoittaa, että sen liike on

riippuvainen alkuperäisestä luusta. (Totten 2012, 160) Näin saadaan luotua luuketjuja, jotka toimivat kuten esimerkiksi ihmisen luuranko toimii. Jos esimerkiksi selkäranka kääntyy, monet muut luut kääntyvät automaattisesti samanaikaisesti.

Näiden riippuvuuksien lisäksi luille on mahdollista asettaa erilaisia rajoitteita. Näistä tärkein on IK eli inverse kinematics, joka mahdollistaa luuketjun viimeisen luun hallita koko luuketjua. (Totten 2012, 161) Tästä hyvänä esimerkkinä voisi toimia ihmisen jalka. Kun animaattori liikuttaa jalkaterää, koko muu jalka seuraa luonnollisesti perässä.

OWLS-pelin mäyrähahmoa rigatessa käytettiin Rigify-lisäosan Wolf-meta-rigiä. Tämä sudelle tarkoitettu riggaus toimi varsin hyvin nelijalkaiselle nisäkkäälle, mäyrälle. Kun suden animaatioluuranko oli luotu, sitä piti muokata siten, että luiden pituudet ja paikat vastasivat 3D-mallinnettua mäyrää. Erityisesti raajojen, pään ja hännän luut tuli asettaa niin, että ne vastaavat täysin hahmon mittasuhteita. Tämä takasi animaatioiden tekemisen sujuvuuden. Riggauksessa on animaatioiden kannalta myös tärkeää, että hahmon ruumiinosien suhteellinen paino on ainakin lähellä oikeanlaista. Weight painting on tekniikka, jolla voidaan varmistaa, että hahmon eri osat reagoivat animaatioluurangon liikkeeseen oikealla tavalla. Näin saadaan luotua vaikutelma, että esimerkiksi vaate voi heilua liikkeen aikana, mutta eläimen pää ei voi luonnollisesti venyä ja muuttaa muotoaan rajattomasti. Kun rigify-luuranko on muokattu halutun muotoiseksi, se voidaan liittää animoitavaan hahmoon. Hahmosta tehdään alisteinen luurangolle, jolloin hahmo liikkuu luurangon mukana. Blenderissä on ominaisuus ”Parent with automatic weights”, joka nimensä mukaisesti tekee edellä mainitun weight paintingin automaattisesti. Lopputulos ei ole aina täydellinen ja käsityötä saatetaan tarvita, mutta ominaisuus helpottaa ja nopeuttaa työtä huomattavasti. Lopuksi luurangosta voidaan vielä Rigifyn avulla tehdä hahmolle kontrollit, joiden avulla animoinnin voi helpommin toteuttaa. Kuvassa 22 nähdään mäyrän animaatiokontrollit.



KUVA 22. Rigify-kontrollit. Mäyrä on valmis animoitavaksi.

### 3.2.2 Animointi

On hyvä käsitellä animaation tekemisen peruskulku, jotta animaatioprosessi olisi alusta asti selkeä. Käytetään esimerkkinä nimenomaan kävelyanimaatiota, sillä se on jokaisen animaattorin hallittava. Animaation voi luoda kahdella päätävällä: suorana animaationa tai asennosta asentoon (Toten 2012, 161). Suorassa animaatiossa animaattori tekee jokaisen liikesarjan kuvan erikseen. Jälkimmäinen on huomattavasti helpompi tapa tehdä liikesarjoja peliä varten. Tällöin 3D-animaattori tekee tietyn määrän avainkuvia, eli asettaa hahmon animaatioluurangon avulla eri asentoihin eri kohdissa animaation aikajanaa. Animaatio-ohjelma, tässä tapauksessa Blender, tekee sitten automaattisesti liikkeen kahden avainkuvan välille luoden animaation avainkuvien väliset kuvat. Avainkuvista nähdään esimerkki kuvassa 23.



KUVA 23. Esimerkki kävelyanimaation avainkuvista.

Kun tehdään kävelyanimaatiota, animaation tulee olla itseään toistava. Eli animaation loputtua sen tulee alkaa saumattomasti alusta, jotta saadaan aikaan vaikutelma jatkuvasta kävelystä. Näin ollen animaation ensimmäisen ja viimeisen avainkuvan tulee olla identtiset. Animaation pituus asetetaan kuitenkin yhtä kuvaa lyhyemmäksi kuin animaation pituus, jolloin viimeistä avainkuvaa ei näytetä, mutta liike säilyy.

Kun liikesarjan animaatio on saatu valmiiksi 3D-ohjelmassa, se täytyy renderoida. Prosessi ei sinänsä eroa jo aiemmin tässä työssä kuvatusta renderointiprosessista. Nyt vaan renderoidaan animaatiosta usean kuvan kuvasarja yhden kuvan sijaan. Kun kuvasarja on saatu renderoitua, animaatiot pitää vielä tuoda 2D-ympäristöön. Seuraavassa luvussa käydään läpi 2D-animaatioiden luomista peliä varten.

### **3.3 3D-animaatiot sprite-animaatioiksi**

Jos 3D-animaatioista renderoidut kuvasarjat halutaan tuoda 2D-peliin, niistä pitää tehdä niin sanotut spritesheetit. Spritesheet on yksi kuva, johon on sisällytetty yhden tai useamman liikesarjan kaikki kuvat, jotka liikkeen näyttämiseen tarvitaan. Tällä tavalla aikaansaatu liikesarjaa kutsutaan sprite-animaatioksi. OWL-Sin tapauksessa esimerkiksi kävelyanimaatiossa oli 63 kuvaa, mutta niitä kaikkia ei olisi ollut järkevä käyttää, sillä spritesheetistä olisi tullut valtava. Näin ollen valittiin 10 kuvaa, joiden avulla liikkeestä saatiin riittävän sulava, mutta spritesheetin koko ei kasvanut liian suureksi.

Spritesheetiä tehdessä kuvat tuodaan Smart Objecteina Photoshopiin, jossa ne muokataan peliin sopivaan kokoon. Jokaista kuvaa kannattaa muokata yhtä aikaa, sillä niiden täytyy vastata suhteelliselta kooltaan alkuperäistä animaatiota. Muokkauksen jälkeen kuvat järjestetään peräkkäin kuva-alueelle vasemmalta oikealle. Kuvia siirrellessä kannattaa muistaa, ettei kuvia saa liikuttaa pystysuunnassa suhteessa toisiinsa, koska tällöin 3D-animaatiossa saavutettu liike turmeluu. Kun kuvat ovat järjestyksessä, spritesheet tallennetaan png-formaatissa ja

se on valmis käytettäväksi pelimoottorissa. Samalla tavalla kuin kävelyanimaatio, voidaan kaikki muutkin 3D-animaatiot muuttaa 2D-peliin sopiviksi. 3D-animaatioista valikoiduista kuvista tehdään siis kuvasarjoja, jotka järjestetään kuvankäsittelyohjelmassa spritesheeteiksi. Kuvat 24 ja 25 ovat esimerkkejä pelihahmon spritesheeteistä.



KUVA 24. Esimerkki kävelyanimaation spritesheetistä.

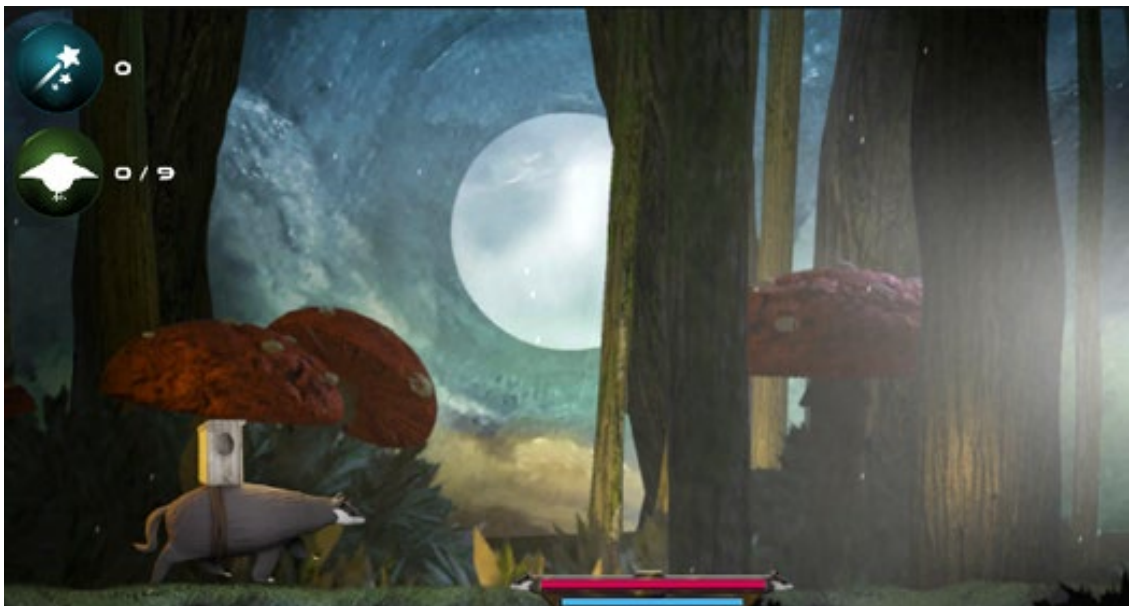


KUVA 25. Esimerkki lentoanimaation spritesheetistä.

## 4 EFEKTIT

### 4.1 Efektointi osana peliympäristöä

Erilaiset erikoiseffektit, esimerkiksi Unityn partikkeliefektit, elävöittävät entisestään huolella tehtyä peliympäristöä. Effektit eivät sinänsä kuulu tämän opinnäytteen varsinaiseen sisältöön, mutta ne ovat peliympäristön visuaalisena elementtinä hyvä mainita lyhyesti. Partikkeliefektien avulla voidaan luoda peliympäristöön esimerkiksi sääilmiöitä, taikuutta tai muita vastaavia tunnelmaa luovia elementtejä, jotka artisti kokee osaksi peliympäristöä. OWLS-peliin luotiin erilaisia sääilmiöitä, kuten sadetta, usvaa, suosta nousevaa myrkköhöyryä, revontulia, lumisadetta, tuulessa lentäviä höyheniä ja niin edespäin. Pelin ohjelmoija loi efekteille vielä järjestelmän, joka tekee sääilmiöistä satunnaisia. Näin voitiin luoda peliympäristöstä entistä elävämpi ja tunnelmallisempi. Kuvassa 26 nähdään peliympäristöön lisättyä sadetta ja usvaa.



KUVA 26. ruutukaappaus pelistä OWLS. Usvaa ja sadetta metsässä.

## 5 POHDINTA

Opinnäytetyössä tarkasteltiin artistin näkökulmasta, kuinka luodaan kaksiulotteista peligrafiikkaa käyttäen sekä perinteistä maalaustekniikkaa että 3D-mallinnusta. Työssä sivuttiin myös grafiikan animointia. Tarkasteltuja keinoja sovellettiin käytännössä opinnäytetyön aikana tehtyyn OWLS-mobiilipeliin. Voidaan todeta, että opinnäytetyössä läpikäytyt työtavat 3D-grafiikasta maalaukseen soveltuvat erinomaisesti 2D-grafiikkaa luodessa. Erityisesti 3D-tekniikan etuna on valaistuksen mahdollisuudet ja sen myötä tunnelman luominen peliympäristöihin. Valot ja varjot on helppo toteuttaa ja niistä saadaan tarvittaessa realistiset ja loogiset. Perinteisillä maalaustekniikoilla voidaan tehdä yksilöllistä grafiikkaa, jonka avulla pelin visuaalinen identiteetti erottuu helpommin massasta. Opinnäytetyössä saatiin luotua hyvät suuntaviivat artistin työnkulkuun edellä mainittuja tekniikoita käyttäen. On selvää, että peliartistin on hyvä hallita monia eri tekniikoita grafiikan luomiseen, sillä tämä antaa edellytykset persoonalliselle ja huolitellulle visuaaliselle ilmeelle.

Jos pelituotannossa käytettäisiin enemmän perinteisiä maalaus- ja piirustustekniikoita, se voisi osaltaan madaltaa peliprojektien kustannuksia. Tämä vaatisi kuitenkin toimenpiteitä myös esimerkiksi peliartistien koulutuksessa sekä suuria muutoksia artistien työnkulkuun. Perinteisiä maalaustekniikoita ei esimerkiksi TAMKin pelituotannon opintosuunnitelmiin nykyisellään kuulu, mikä osaltaan ohjaa peliartisteja ainoastaan digitaalisiin työtapoihin. Jos pelikoulutukset tarjoaisivat opiskelijoille valmiudet perinteisten maalaustekniikoiden käyttämiseen pelituotannossa, myös peliyrityksien olisi helpompi löytää osaajia ja pohtia omia työnkulujaan tästä näkökulmasta. Yhdistämällä eri tekniikoita peligrafiikan luomisessa voisi olla mahdollista kehittää nopeita ja kustannustehokkaita työnkuluja peliprojektista riippuen. Pitää kuitenkin muistaa, että vaikka rahaa säästyisi, ei perinteiset maalaustekniikat välttämättä sovi kaikkien pelien ulkoasuun. Artistilla on kuitenkin vastuu luoda peliin visuaalinen ilme, joka on relevantti juuri sille pelille. Pelillä tulee olla ulkoasu, joka luo oikeanlaisen tunnelman ja kontekstin.

## LÄHTEET

Ahearn, L. 2008. 3D game environments : create professional 3D game world.

Bone S. 2014. Parallax Scrolling: A Simple, Effective Way to Add Depth to a 2D Game. Luettu 7.1.2022.

<https://gamedevelopment.tutsplus.com/tutorials/parallax-scrolling-a-simple-effective-way-to-add-depth-to-a-2d-game--cms-21510>

Cearley, C. 2020. This Art Advice Will Save You Years!-video.

Julkaistu 23.5. 2020. Viitattu 12.1.2022.

<https://www.youtube.com/watch?v=ni4ts22XFsw>

Dark Souls - Boss: Four Kings. Kuvakaappaus YouTube-käyttäjän Mark Zoltonin Dark Souls -pelaamisesta. YouTube 2011. Katsottu 4.1.2022.

<https://www.youtube.com/watch?v=fg1A31pReN8>

DoubleFineProd 2016. The Making of Day of the Tentacle Remastered. YouTube-video. Julkaistu 2016. Viitattu 5.1.2022.

<https://www.youtube.com/watch?v=LjF4eMrYfG0>

Morrison, M. 2002. Sams Teach Yourself Game Programming in 24 Hours

Plourde, P. 2014. The Art of Child of Light. GDC Vault -video.

Julkaistu 2014. Viitattu 5.1.2022.

<https://www.gdcvault.com/play/1020561/The-Art-of-Child-of>

Powell, A.W. 2010. Blender 2.5 lighting and rendering bring your 3D world to life with lighting, compositing, and rendering.

Renderguide.com 2022. Blender Eevee vs Cycles. Luettu 7.1.2022.

<https://renderguide.com/blender-eevee-vs-cycles-tutorial/>

Sloan, R. 2015. Virtual character design for games and interactive media.

Totten, C. 2012. Game Character Creation with Blender and Unity.

Vasconcelos, V. 2011. Blender 2.5 character animation cookbook: 50 great recipes for giving soul to your characters by building high-quality rigs and understanding the principles of movement.

Williams, G. 2016. On average, how fast are concept artists expected to complete a design? Quora.com question answer. Read 12.1.2022

<https://www.quora.com/On-average-how-fast-are-concept-artists-expected-to-complete-a-design>

Willmore, B. 2004. Adobe® Photoshop® CS Studio Techniques.

Wolf, M.J.P. 2007. The Video Game Explosion: a History from PONG to PlayStation and Beyond : A History from PONG to PlayStation and Beyond.