Mira Kajander

Animaatiosyklien tekeminen peliympäristöön

3D-animaatio peleissä ja sen luominen

Metropolia Ammattikorkeakoulu

Medianomi (AMK)

Viestintä

Opinnäytetyö

Päivämäärä

Tiivistelmä

Tekijä(t):	Mira Kajander
Otsikko:	Animaatiosyklien tekeminen peliympäristöön
Sivumäärä:	42 sivua + 1 liitettä
Aika:	19.5.2020
Tutkinto:	Medianomi (AMK)
Tutkinto-ohjelma:	Viestintä
Suuntautumisvaihtoehto:	3D-animointi ja -visualisointi
Ohjaaja(t):	Lehtori Peke Huuhtanen

Tässä opinnäytetyössä käydään läpi prosessi, jossa peliympäristöön tarkoitetulle 3D-hahmolle luodaan animaatiosyklejä, jotka toimivat pelimoottorissa ja ovat käytännössä valmiita lisättäväksi peliin. Työssä käsitellään aluksi 3D-animaation prosessia ja historiaa, sekä 2D- ja 3D-animaation välisiä eroja. Myöhemmin käsitellään myös sitä, millaisia vaatimuksia animaatiolla on peliympäristössä, ja miten pelin visuaalinen tyyli vaikuttaa animaatioon.

Työhön kuuluu myös projekti, jossa kettuhahmolle luodaan animaatiosyklejä ja niitä testataan pelimoottorissa. Työssä käydään läpi eri syklien animoimisen prosessia ja kerrotaan animaation valmistelusta ja siirtämisestä pelimoottoriin. Kettuhahmo on peliyritys Frozenbyten omaisuutta, jonka projektiin nämä animaatiot tehtiin.

Animaatiosyklien luominen vaatii kykyä rytmittää animaatio, projektiin valitun pelimoottorin käytön ja sen standardien oppimista sekä ymmärrystä siitä, miten eri animaatiot saadaan peliympäristössä sulautumaan liikkeestä toiseen halutulla tavalla.

Avainsanat: 3D, 2D, animaatio

Abstract

Author(s):	Mira Kajander
Title:	Creating animation cycles into gaming environment
Number of Pages:	42 pages + 1 appendices
Date:	19 May 2021
Degree:	Bachelor of Media
Degree Programme:	Media Communication
Specialisation option:	3D Animation and Visualization
Instructor(s):	Peke Huuhtanen, Senior Lecturer

The purpose of this thesis is to go through the process of creating animation cycles for a 3D character, with the intent to test the animations in a game engine, ready to be used in an actual game. First the thesis goes through the history of 3D animation as well as the process of creating 3D animations and the differences between 2D and 3D animations.

This thesis also includes a project where various animation cycles will be created for a fox character, which will then be tested in a game engine. At first the work will go through the process of creating the animation cycles and preparing them for the game engine, where the animations will be tested. The fox character belongs to game company Frozenbyte, who’s project these animations were made for.

The creation of animation cycles requires ability to give the animations a pleasant rhythm, proper knowledge of the game engine the project will be using, and understanding how to blend two animations together in a game environment.

Keywords: 3D, 2D, animation

Sisällys

1 Johdanto 1

2 Mitä 3D-animaatio on 1

2.1 3D-animaation historiaa 2
2.2 Eroavaisuudet 2D- ja 3D-animaatioiden välillä 6
2.3 3D-animaation prosessi 8
2.3.1 Mallintaminen 9
2.3.2 Sommittelu ja animointi 11
2.3.3 Renderöinti 12

3 Animaatio peliympäristössä 13

3.1 Videopelianimaation vaatimukset verrattuna elokuvatuotantoon 13
3.1.1 Kamera 13
3.1.2 Liikkeiden variaatiot ja blendaus 14
3.1.3 Ohjelmointi 15
3.1.4 Interaktiivisuus 17
3.2 Pelin visuaalisen tyylin vaikutus animaatioon 17
3.2.1 Dead by Daylight 17
3.2.2 Untitled Goose Game 18
3.2.3 Minecraft 19

4 Animaatiosyklien prosessi kettuhahmolla 20

4.1 Tutkimustyö ja referenssien etsiminen 20
4.2 Syklien animointi 21
4.2.1 Seisominen 22
4.2.2 Ympärille katselu 22
4.2.3 Nuuskiminen 26
4.2.4 Kävely 29
4.2.5 Juoksu 32
4.3 Animaatioiden lisääminen pelimoottoriin 33
4.3.1 Tiedostojen valmistelu 33
4.3.2 Animaatiokontekstien luominen 34
4.3.3 Pelimoottorissa testaaminen 35

5 Pohdinta 36

Lähteet 37

5.1 Kuvalähteet 38

Liitteet 40

Ketun valmiit animaatiosyklit 40

1 Johdanto

Tämän opinnäytetyön tavoitteena on käydä läpi prosessi, jossa peliympäristöön tarkoitetulle 3D-mallinnetulle hahmolle luodaan animaatiosyklejä, jotka toimivat pelimoottorissa ja jotka ovat käytännössä valmiita lisättäväksi peliin.

Teoriaosuudessa selvennetään, mitä 3D-animaatio on, sekä käydään läpi sen historiaa ja sitä, miten pelianimaatio eroaa esimerkiksi elokuvien animaatiosta. Projektiosuudessa käydään läpi animaatiosyklien luomisprosessin kettuhahmoa käyttäen. Tämä prosessi sisältää referenssien keräämisen ja tutkimustyön, syklien animoinnin sekä animaatioiden viemisen pelimoottoriin ja niiden testaamisen pelimoottorin sisällä.

3D-animaatio on kehittynyt vuosien saatossa suunnattomasti, ja nykyään CGI-animaatio eli tietokoneanimaatio on standardi niin elokuvissa, televisioissa kuin peleissä. Sitä käytetään kuitenkin myös esimerkiksi markkinoinnissa ja lääketieteessä. Videopelien grafiikat ja animaatiot kehittyvät vuosi vuodelta enemmän konsolien kyetessä pyörittämään yhä raskaampia ja vaativampia grafiikoita ja 3D-työkalujen parantuessa.

Peliala on yhä kasvava ala niin Suomessa kuin muualla maailmassa, ja pelien parissa työskentely on monelle nuorelle haave, ja mielenkiinto videopelien teon prosessia kohtaan kasvaa pelialan mukana. (Autio & Hautala 2020.)

2 Mitä 3D-animaatio on

3D-animaatio on tietokonegeneroitujen objektien tai hahmojen liikuttamista kolmiulotteisessa ympäristössä. Animoitavat objektit voivat liikkua samoja periaatteita seuraten kuin oikeassa elämässä, tai ne voidaan animoida vapaammin ja tyylitellymmin. Yleensä graafisen tyylin realismin määrä määrittelee myös animaation tyylin. Jos hahmot ovat realistisia, niin on animaatiokin. Jos hahmot ovat

yksinkertaisia ja karikatyyrimäisiä, myös animaatiot voivat olla erittäin liioiteltuja ja tyyliteltyjä.

Nykyään 3D-animaation käyttötarkoitukset ovat laajat, ja niitä käytetään esimerkiksi elokuvissa, tv-sarjoissa ja peleissä, mutta myös mainoksissa ja markkinoinnissa ja jopa lääketieteessä. (Beegel, 2020.)

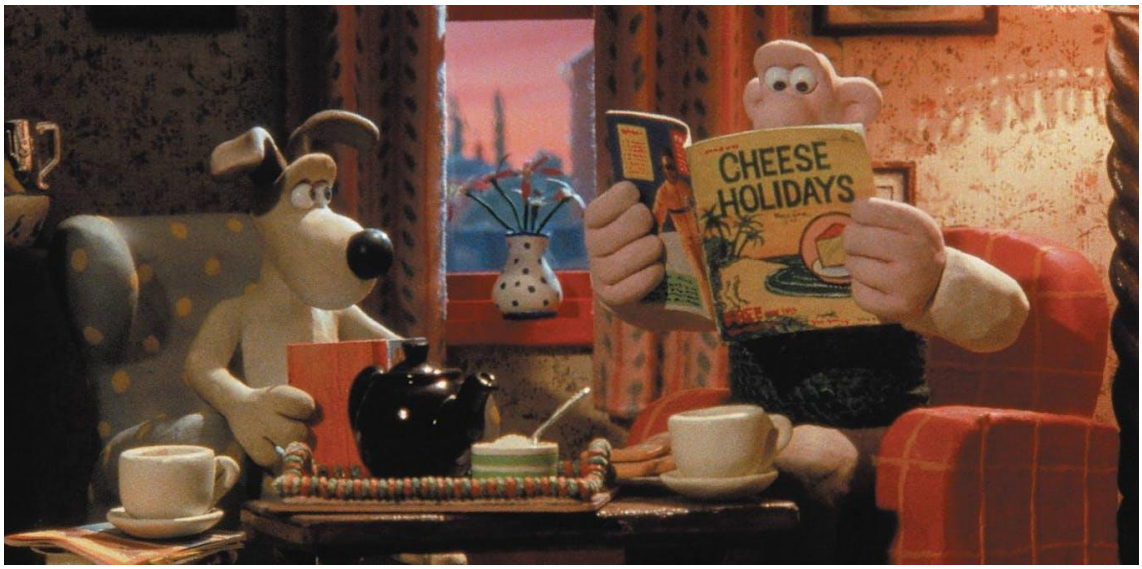
Yksi esimerkki 3D-animaatiosta ja grafiikasta on vuonna 2018 julkaistu Monster Hunter World -peli (kuva 1). Kyseinen peli on esimerkki realistisemmasta graafisesta tyylistä, vaikka se sisältää erittäin vahvoja fantasiaelementtejä.

Kuva 1. Monster Hunter World (Pramath, 2017)

2.1 3D-animaation historiaa

3D-animaatio sai alkunsa animaatiotekniikan kautta, jota kutsutaan stop-motion tai claymation-animaatioksi. Tässä tekniikassa käytetään joko nukkeja tai muovailumassasta tehtyjä hahmoja. Näistä nukeista otetaan kuvia, joiden välissä hahmon asentoa vaihdetaan hieman kerrallaan. Kun otetut kuvat toistetaan peräkkäin, hahmo näyttää liikkuvan. Näitä tekniikoita on käytetty jopa 60-luvulta lähtien. (Animation Academy 2011.)

Yksi tunnetuimpia esimerkkejä claymation-tekniikasta on Wallace ja Gromit -elokuvat (kuva 2). Vuonna 2016 julkaistu Kubo ja samuraiseikkailu on taas moderni esimerkki stop-motion-tekniikasta (kuva 3).

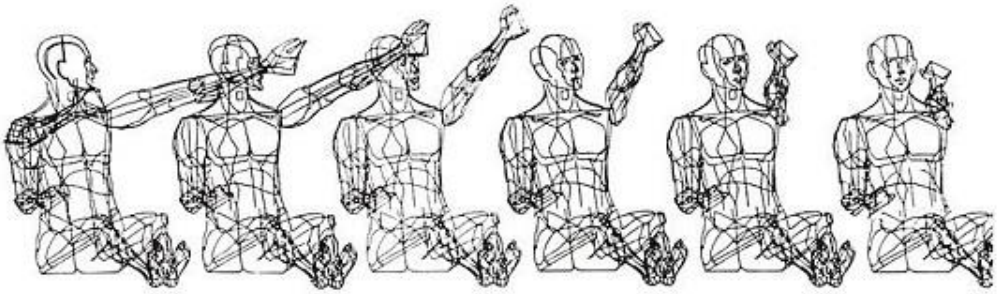


Kuva 2. Wallace ja Gromit (Allen 2017)

Kuva 3. Kubo ja samuraiseikkailu (Pile 2019)

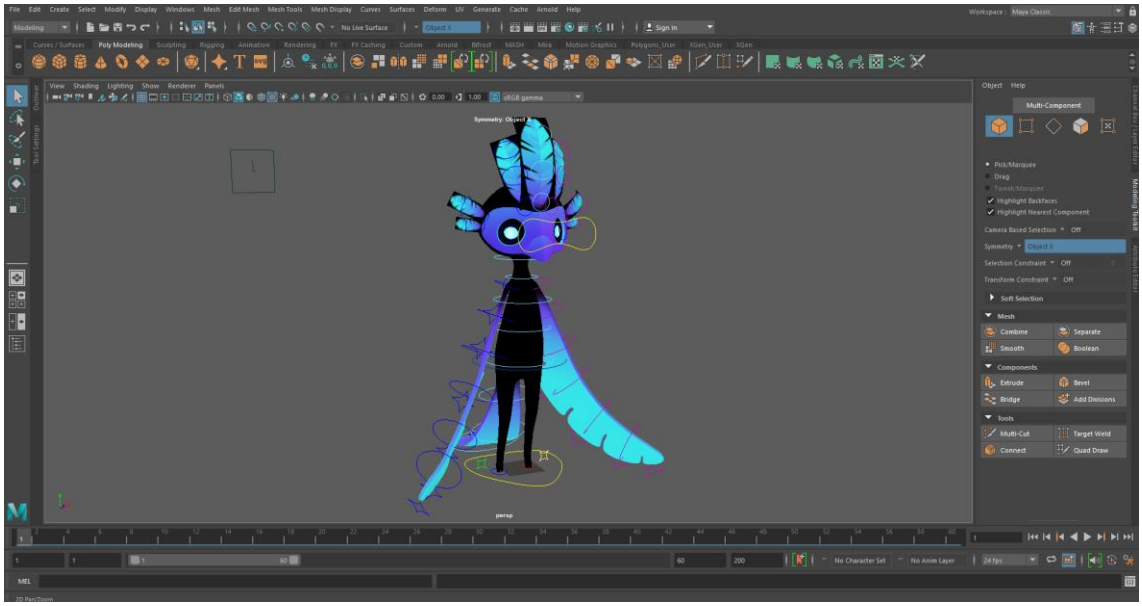
Vuonna 1964 William Fetter, joka tähän aikaan oli töissä Boeing-yhtiöllä, loi ensimmäisen kolmiulotteisen kuvan ihmiskehosta tietokonegrafiikoiden avulla. Malli

luotiin käytettäväksi lentokoneiden ohjaamon suunnittelun apuna ja siihen liittyvän tutkimusdatan keräämiseen. Tälle 3D-kuvalle annettiin nimi ”Boeing Man” eli Boeing-mies (kuva 4). (Norman 2007.)



Kuva 4. Boeing Man (Introbrand 2018)

Vuonna 1982 John Walker perusti Autodesk-yrityksen, jonka ensimmäisiä tuotteita oli AutoCAD, joka on tietokoneavusteisen suunnittelun ohjelmisto (Plantec 2012). Nykyään Autodeskin ohjelmistot kuten Maya (kuva 5) ja 3ds Max ovat standardeja monissa 3D-alan yrityksissä.



Kuva 5. Kuvakaappaus Maya-ohjelmasta

1980-luvulla Turner Whitted kehitti ray tracing -renderöintitekniikan, joka mahdollisti realistisemman lopputuloksen animaatiolle. Wavefront Technologies taas julkaisi ensimmäisen 3D-animaatio-ohjelman. (Introbrand 2018.)

Pixar oli ensimmäinen animaatiostudio, joka alkoi käyttää suuria määriä 3D-animaatiota elokuvissaan. Vuonna 1995 Pixar julkaisi ensimmäisen kokonaan tietokoneella animoidun elokuvan, Toy Storyn. Elokuva oli tuolloin teknisesti uraauurtava sisältäen monimutkaisia hahmoja, tekstuureja ja animaatioita. (Animation Academy 2011.)

1990-luvulla 3D-animaation menestys kasvoi ja sitä alettiin hyödyntää enemmän ja enemmän Hollywood-tuotannoissa (Introbrand 2018). Myös peleissä käytetty 3D-tuotanto alkoi kehittyä. Vuonna 1994 Playstation-konsoli julkaistiin Japanissa ja vuotta myöhemmin Amerikassa. Konsoli oli kriitikoiden ylistämä ja myös taloudellinen menestys. Yksi konsolin tunnetuimpia pelejä on Crash Bandicoot (kuva 6).

Kuva 6. Crash Bandicoot (Kellett 2016)

Kuva 7. Super Mario 64 (Smithers 2020)

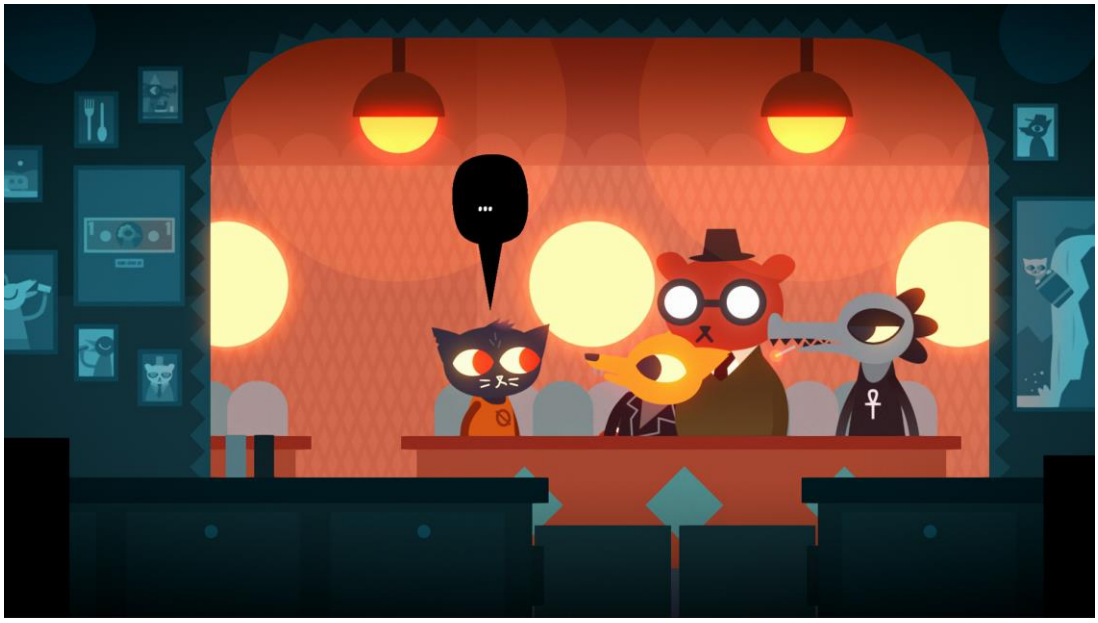
Vuonna 1996 Nintendo julkaisi Nintendo 64 -pelikonsolin, joka oli yhtiön kolmas konsoli, ja ensimmäinen, joka hyödynsi peleissään 3D-grafiikkaa. Konsolin ohella julkaistiin Super Mario 64 (kuva 7), ensimmäinen kolmiulotteinen Super Mario-peli. Pelaajat pystyivät liikuttamaan hahmoa kolmiulotteisessa ympäristössä, mikä oli siihen aikaan peleissä harvinaista. (Fernandes Da Conceição 2019.)

2.2 Eroavaisuudet 2D- ja 3D-animaatioiden välillä

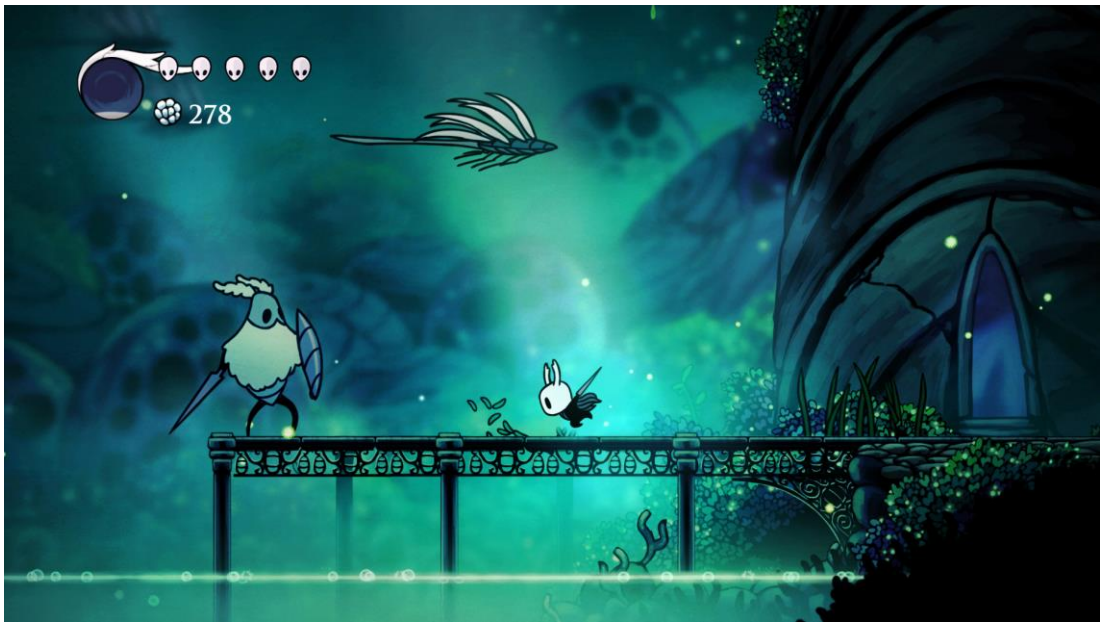
2D- ja 3D-animaatio eroavat toisistaan niin ulkonäöllisesti kuin teknisesti. 2D-animaatiolla tarkoitetaan animaatiota, joka on luotu yksittäisistä piirretyistä kuvista, jotka muodostavat liikettä, kun ne toistetaan peräkkäin.

2D-animaatiossa hahmot ovat nimensä mukaan kaksiulotteisia ja muodostuvat kahdesta akselista, pituudesta ja leveydestä. 3D-hahmolla on näiden lisäksi myös kolmas akseli, syvyys (Pixelloid Studios, 2018). 3D-hahmoa voi helposti animoida pyörimään 360 astetta, kun taas 2D-animaatiossa jokainen kuva tulisi piirtää erikseen, mikä tekisi tästä erittäin työlästä.

Yleisesti ottaen 2D-animaatio on kuitenkin halvempaa ja nopeampaa tuottaa kuin 3D-animaatio, etenkin pelejä tehdessä. Siksi useat indie-pelit, eli pelit, joita on tekemässä pieni tiimi pienellä budjetilla, toteutetaan käyttäen 2D-animaatiota. Esimerkkejä tällaisista peleistä ovat Night in the Woods (kuva 8) ja Hollow Knight (kuva 9).



Kuva 8. Night in the Woods (Infinite Fall 2017)



Kuva 9. Hollow Knight (Team Cherry 2017)

Designblendz Team (2019) tuo artikkelissaan esille näiden 2D- ja 3D-tekniikan eroavaisuuksia. 2D-animaatiossa animaattorin tarvitsee piirtää hahmo vain siitä kuvakulmasta, joka katsojan on tarkoitus nähdä, ja vain ne osat, jotka kuvassa sillä hetkellä näkee. 3D-animaattori taas liikuttaa kokonaista, valmista hahmoa. Hahmolle on luotu kontrolleja, jotka ovat kytköksissä hahmolle luotuun luurankoon. Näitä kontrolleja liikuttamalla animaattori pystyy asettelemaan ja animoimaan hahmoa. Designblendz Team (2019) käyttää vertausta, jossa 3D-animaattori on kuin nukkemestari, joka liikuttelee nukkeaan kolmiulotteisessa tilassa. (Designblendz Team 2019.)

3D-animaatiossa animaattorilla on täysi kontrolli kameran käyttäytymisestä. Kohtaukset, jotka perinteisessä 2D-animaatiossa olisivat teknisesti hyvin haastavia, ovat yksinkertaisia toteuttaa tietokoneanimaatiolla. Esimerkiksi jos takaapäin kuvattu hahmo halutaankin muuttaa niin, että tätä kuvattaisiin sen sijaan edestäpäin, 3D-animaatiossa tarvitsee vain siirtää kameran paikkaa. 2D-animaatiossa kuvatun kulman vaihto tarkoittaa usein animaation kokonaan alusta aloittamista.

2D-animaatio luodaan piirtämällä yksittäisiä kuvia, freimejä, jotka eroavat hieman toisistaan. Kun nämä kuvat toistetaan peräkkäin, liikkeen tuntu syntyy. 3D-animaatiossa animaattorin ei tarvitse animoida hahmon jokaista freimiä erikseen. Animaattori määrittelee tärkeimmät asennot, ja kone laskee liikeradan näiden kahden asennon välille. Tämän jälkeen animaattori hienosäätää ja muokkaa koneen luomaa liikettä, jotta jälki on enemmän halutun näköinen.

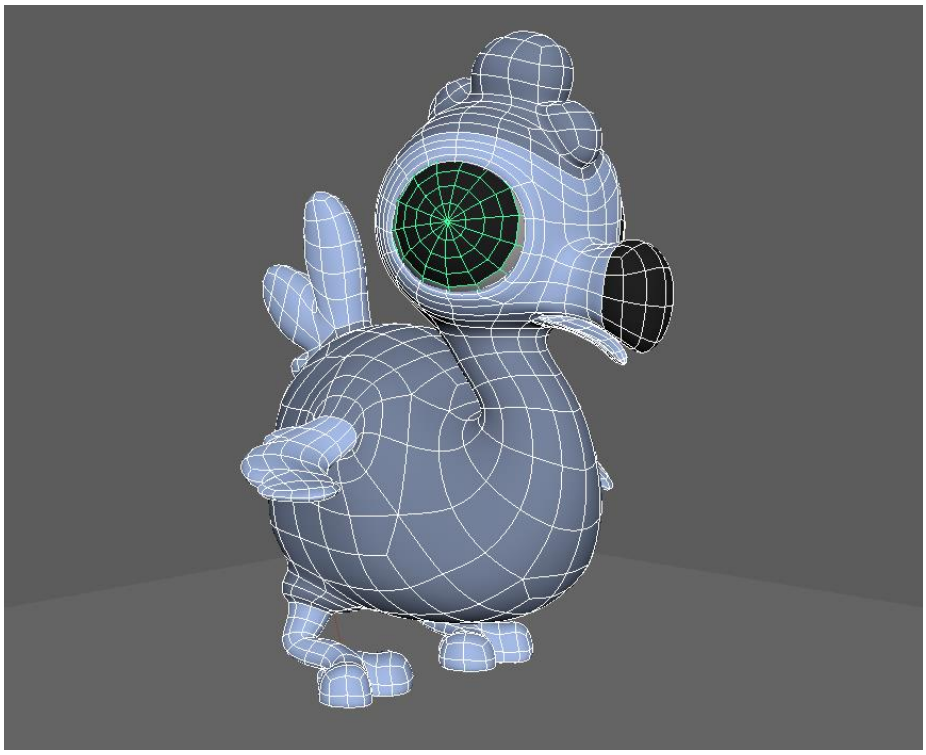
2.3 3D-animaation prosessi

Beetle (2020) kertoo, miten 3D-animaation prosessissa on yleisesti ottaen kolme vaihetta: mallintaminen, sommittelu ja animaatio sekä viimeisenä renderöinti. Riippuen tuotannon tyylistä, esimerkiksi suunnitellaanko animaatiota peliin vai elokuvaan, prosessi on hieman erilainen.

Kaikilla animaatio- tai pelifirmoilla on omat käytäntönsä, mutta perusperiaatteet ovat kaikkialla käytännössä samat.

2.3.1 Mallintaminen

Mallinnuksella tarkoitetaan prosessia, jossa artisti luo 3D-ohjelman avulla animoitavan hahmon tai objektin, jota animaatiossa tullaan käyttämään. Artistin kokonaan itse mallintaman objektin sijasta on nykyään myös mahdollista skannata niitä oikeasta elämästä, jonka jälkeen artisti voi muokata tätä 3D-skannausta animaation tarpeiden mukaiseksi. (PointInTimeStudios 2020.) 3D-skannatut mallit vaativat lähes aina viimeistelyä graafikon toimesta, jotta se on käyttökelpoinen animaatiota varten. Tämä johtuu siitä, että mallinnetulla hahmolla on oltava siisti ja standardien mukainen topologia. Topologialla tarkoitetaan käytännössä hahmon rautalankamallia (kuva 10). Tämä rautalankamalli taas muodostuu sadoista tai tuhansista polygoneista, pienistä monikulmioista. Nämä polygonit määrittävät, kuinka paljon hahmolla on taipumis- tai liikevaraa kussakin kehon kohdassa.



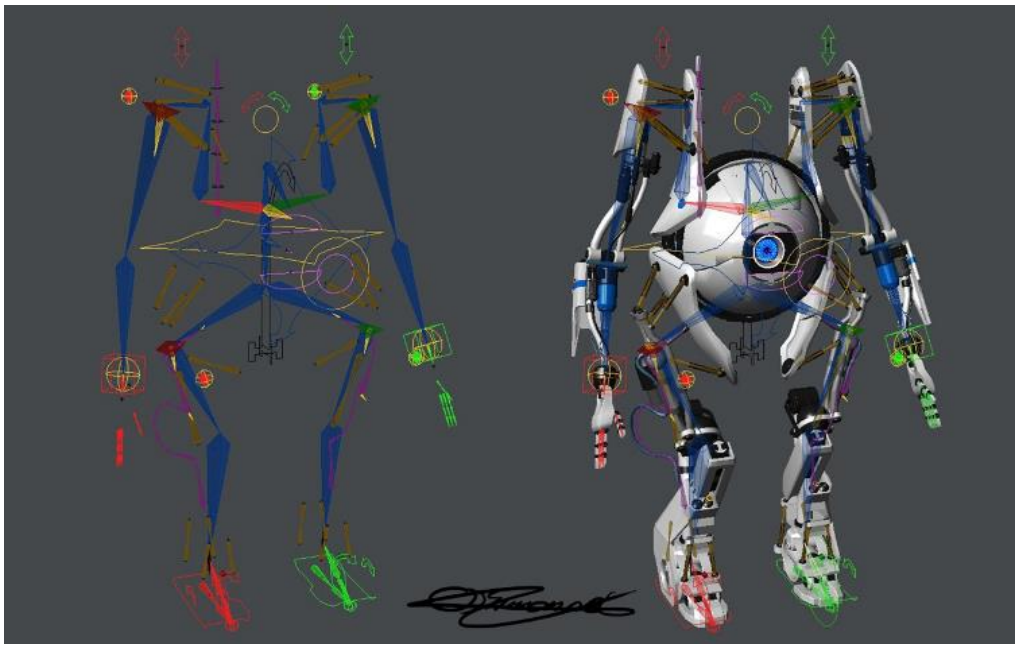
Kuva 10. Dodo-hahmon topologiaa (kuvakaappaus omasta projektista)

Jotta hahmoa saadaan liikutettua sulavasti, ilman epätoivottuja komplikaatioita, on polygonien oltava mielellään nelikulmioita, ja niiden tulee muodostaa silmukoita, jotta hahmon geometria pystyy venymään animaatioiden mukana halutusti.

Kohdat, joissa esimerkiksi hahmon nivelet taipuvat, vaativat enemmän polygoneja.

Mallinnuksen jälkeen hahmolle lisätään tekstuurit ja materiaalit. Tekstuurin avulla hahmolle lisätään värit ja kuviot, kun taas materiaali antaa ohjelmalle informaation siitä, miten valo käyttäytyy osuessaan hahmoon (Beegel 2020.)

Hahmo myös rigataan, eli sille luodaan ohjelman sisäinen luuranko sekä kontrollit, joiden avulla hahmoa pystyy liikuttamaan. Prosessista on eri variaatiota käytetystä ohjelmasta riippuen, mutta periaate on kuitenkin sama. Ensin hahmolle luodaan luuranko tämän geometriaa mukaillen. Kun luut ovat halutulla paikallaan, niille luodaan kontrollit, joiden avulla animaattori voi liikuttaa esimerkiksi hahmon raajoja vaivattomasti. Lopuksi hahmo skinnataan, eli määritellään mitkä luut vaikuttavat mihin osaan hahmon geometriasta. Esimerkiksi pään luun tulee vaikuttaa ainoastaan hahmon päähän, ei olkapäähän tai rintakehään (kuva 11).



Kuva 11. Portal 2 -pelin hahmo Atlas (Learoyd 2019)

2.3.2 Sommittelu ja animointi

Seuraava vaihe on sommittelu, englanniksi layout, ja animointi. Objektit, hahmot ja mahdollinen kamera asetellaan paikoilleen kohtausta varten (kuva 12). Sommittelun on tarkoitus luoda hahmotelma animaation sisällöstä, ajoituksesta sekä esineiden ja hahmojen välisistä välimatkoista ja niiden kokoeroista.

Kuva 12. 3D-layout Sonic Boom -sarjasta (Andrias 2018)

Sommitteluvaiheen jälkeen lisätään tarvittava animaatio joko puhtaasti animaattorin animoimana tai sitten mocap- eli liikekaappaustekniikkaa hyväksikäyttämällä. Mocap-tekniikkaa käyttäessä näyttelijät pukeutuvat mocap-pukuun, joka sisältää sensoreja eri kohdassa kehoa. Näyttelijä esittää tarvittavat liikkeet, ja puvun sensorien kaappaama animaatiodata siirretään tietokoneelle, josta animaattori pystyy muokkaamaan sitä tarpeen mukaan. (Beegel 2020.)

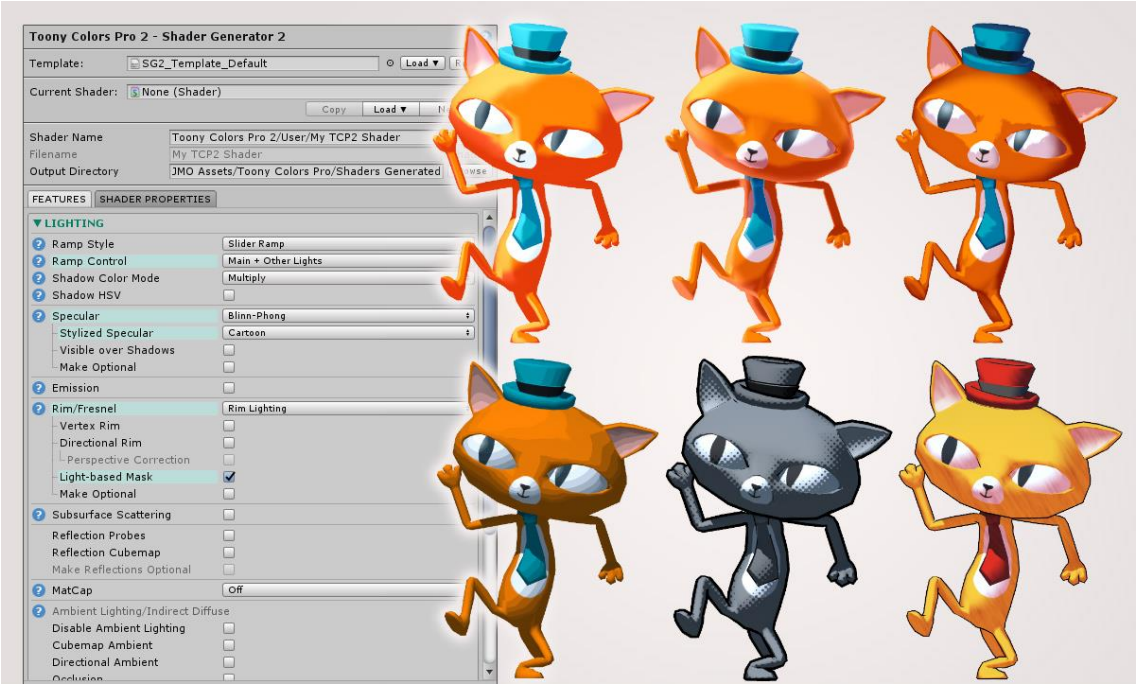
Televisiotuotannossa sommittelu on tärkeä osa animoinnin prosessia, pelituotannossa tämä vaihe on animaattorin puolesta usein tarpeeton, ellei kyse ole elokuvamaisesta välianimaatiosta, jotka ovat ennalta animoituja ja pelaajalla ei ole niihin kontrollia. Näihin välianimaatioihin käytetään sommittelua.

Pelituotannossa sommittelusta vastaa käytännössä kenttäsuunnittelija, ei animaattori, sillä taustan elementtien ja tavaroiden sijainnin on tarkoitus tukea kenttäsuunnittelua ja pelimekaniikkoja. Kenttäsuunnittelussa prioriteettina on pelattavuus ja selkeys.

2.3.3 Renderöinti

Viimeinen vaihe on renderöinti, jolla tarkoitetaan vaihetta, jossa animaatiosta generoidaan valmista kuvaa. Tietokone tulkitsee 3D-tiedoston raakaa dataa, esimerkiksi valoja, varjoja ja heijastuksia, ja käsittelee ne valmiiseen muotoon.

Renderöinnillä on suuri vaikutus lopullisen tuotteen ilmeeseen. Lopputulos voi olla tyyliltään realistista ja oikeaa elämää imitoivaa tai yksinkertaista, vapaata ja taiteellista (kuva 13). (PointInTimeStudios 2020.)



Kuva 13. Useita eri tapoja renderöidä sama hahmo (Moreno 2021)

3 Animaatio peliympäristössä

Peliin tehdyillä grafiikoilla ja animaatioilla on omat tekniset vaatimuksensa, jotka niiden parissa työskentelevän tiimin on otettava huomioon. Kaiken tuotetun materiaalin on toimittava pelimoottorissa, ja animaatioiden on sulauduttava saumattomasti liikkeestä toiseen, jotta pelaajan kokemus ja kontrollin tunne on saumaton. Pelituotannon animaatioita työstäessä vaativin ja tärkein osuus on ehdottomasti pelattavien hahmojen animaatiot.

Tausta- ja sivuhahmoilla, joita pelaaja ei itse kontrolloi, ei välttämättä ole niin tiukkoja vaatimuksia animaatioiden suhteen, koska näiden hahmojen liikkeet eivät reagoi pelaajan käskyihin. Pelituotannossa animaatiot tuodaan peliin ohjelmoinnin ja pelimoottorin avulla, kun taas elokuva- ja televisiotuotannossa lopputulos on videoformaatti.

3.1 Videopelianimaation vaatimukset verrattuna elokuvatuotantoon

Vaikka sekä elokuva- että pelialalla animaattorit käyttävät samoja työkaluja, tekniikoita ja periaatteita, niiden parissa työskentely ei ole kuitenkaan täysin samankaltaista. Televisio- ja elokuvatuotannossa lopputulos on videomuodossa, kun taas pelituotannossa lopputulos on interaktiivinen media.

Videopelit ovat interaktiivista mediaa, jota pelaaja kontrolloi. Elokuvan etenemiseen tai sen hahmojen käytökseen katsoja ei voi vaikuttaa.

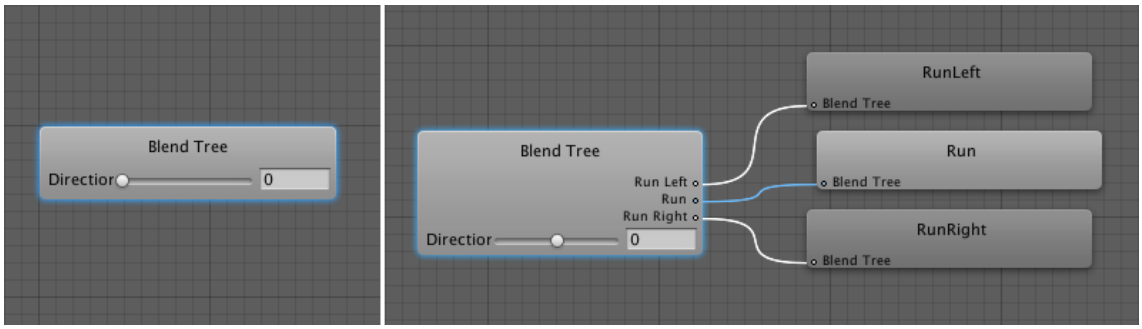
3.1.1 Kamera

Yksi isoimpia eroja elokuva- ja videopelituotannon välillä on kamera. Elokuva-animaatiossa ohjaaja on etukäteen päättänyt, mitä katsoja tulee näkemään. Katsoja näkee animaatiosta vain sen, mitä kameraan on määritelty, ja animaation tarvitsee näyttää hyvältä vain siitä kuvakulmasta, mistä katsoja sen tulee näkemään. (Pluralsight 2020.)

Pelianimaatiossa sen sijaan suurimman osan animaatioista on näytettävä hyvältä kauttaaltaan, sillä pelaajalla on useimmiten mahdollisuus liikuttaa kameraa. Kamera voi olla täysin pelaajan kontrolloitavissa, kuten esimerkiksi Dead by Daylight -pelissä. Pelaaja voi ohjata kameran mihin suuntaan tahansa ja näkee myös oman pelihahmonsa mistä kuvakulmasta tahansa. Apex Legends -pelissä kameran sijainti on aseteltu pelatun hahmon näkökulmaan, eikä pelaaja näe kontrolloimaansa hahmoa. Kamera toimii käytännössä pelihahmon silminä, ja käyttää kameraa kontrolloidakseen, mihin suuntaan hahmo katsoo. Joissain peleissä käytetään täysin staattista kameraa. Crash Bandicoot -peleissä kamera on lukittu pelaajan taakse tai sivulle, eikä sitä voi liikuttaa.

3.1.2 Liikkeiden variaatiot ja blendaus

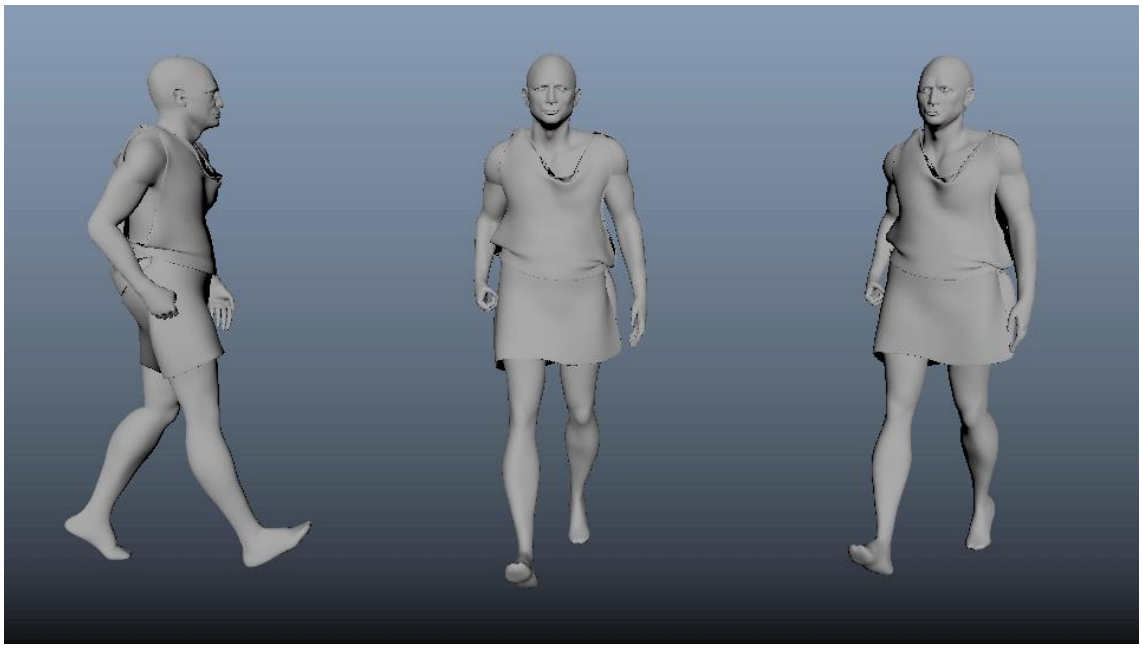
Blendaus tarkoittaa kahden eri animaation siirtymää toisiinsa. Esimerkiksi kun hahmo siirtyy seisomisesta kävelyyn, seisova animaatio blendautuu juoksevaan animaatioon. Elokuvatuotannossa blendausta ei käytetä, vaan animaattorit animoivat liikkeiden siirtymät kokonaan itse. Pelituotannossa kahden eri liikkeen välisen siirtymän määrittää ohjelmointi ja pelimoottori (kuva 14). Kun pelaaja ohjaa hahmon esimerkiksi hyppäämään, vaihtaa pelimoottori animaatiokontekstin hypystä seisomiseen luoden näiden kahden animaatiosyklin väliin siirtymän.



Kuva 14. Blend Tree -asetukset Unity-pelimoottorissa (Unity 2020)

Elokuvatuotannossa animaatioilla on yleensä vain yksi käyttötarkoitus. Elokuva koostuu kohtauksista, jotka sisältävät yleensä kyseiselle kohtaukselle täysin uniikit animaatiot. Elokuvatuotannossa samaa animaatiota ei käytetä uudestaan lähes koskaan.

Pelituotannossa animaatiot koostuvat lyhyistä animaatiosykleistä (kuva 15), esimerkiksi juoksu ja kävely, joiden on saumattomasti vaihduttava keskenään pelaajan käskystä. Samasta liikkeestä saattaa olla eri variaatioita, mutta pelituotannossa samaa animaatiosykliä toistetaan useasti pelikokemuksen aikana.



Kuva 15. Videopelin kävelysykli (Durham 2021)

Elokuvissa on ennalta määritelty koreografia, mutta peleissä pelaaja päättää mitä hahmo tekee ja miten tämä liikkuu (Pluralsight 2020). Kenttäsuunnittelu voi olla enemmän lineaarista, jolloin pelaajan on tarkoitus kulkea ennalta määritettyä reittiä pitkin kentän alusta sen loppuun. Esimerkki tällaisestä pelityylistä on Inside-peli. Kenttäsuunnittelu voi olla myös avoimempaa, kuten Breath of the Wild -pelissä, jossa pelaajalla on täysi vapaus siitä, minne mennä ja mitä tehdä.

3.1.3 Ohjelmointi

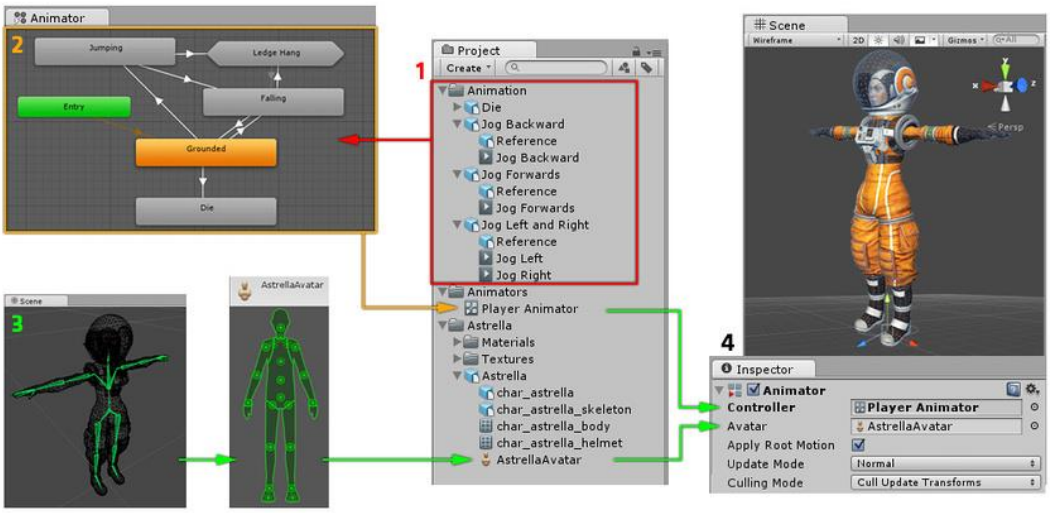
Elokuva-animaatiossa ohjelmointi on tarpeetonta, koska lopputuloksena on videoformaatti. Animaatio luodaan 3D-ohjelmalla ja renderöidään kuvina ulos ohjelmasta. Lopuksi kuvat editoidaan videomuotoon. Videopeleissä hahmot liikku-

vat ohjelmoinnin mukaisesti, ja vaikka animaattorit eivät ole vastuussa pelin varsinaisesta ohjelmoinnista, animaatioita on kuitenkin testattava pelimoottorin sisällä.

Se, mitä animaatioiden siirtäminen pelimoottoriin pitää sisällään, riippuu yrityksen sisäisestä prosessista, mutta käytännössä animaattorin on osattava siirtää ja testata animaatiot pelimoottorin sisällä. Tämä saattaa tarkoittaa esimerkiksi yksinkertaista koodaamista tai vaihtoehtoisesti animaatioiden määrittelyä suoraan pelimoottorin sisällä.

Unity-pelimoottori on esimerkki systeemistä, jossa animaatiot ja niiden ominaisuudet määritellään pelimoottorin sisällä (kuva 16). Yksi näistä ominaisuuksista on Blend Tree. Tämä systeemi sallii useiden eri animaatioiden blendauksen keskenään. Käyttäjä pystyy määrittelemään, kuinka paljon tai vähän mikäkin liike vaikuttaa lopputulokseen. (Unity 2020.)

Kristel (2017) kertoo artikkelissaan esimerkin tilanteesta, jossa Blend Tree -ominaisuuden kyky kustomoida eri liikkeiden vaikutusten voimakkuutta, on hyödyksi. Jos hahmo juoksee eteenpäin mutta samalla kääntyy oikealle, voi animaattori Blend Treen avulla määritellä nämä kaksi animaatiota blendautumaan toisiinsa esimerkiksi sen mukaan, mihin suuntaan hahmo katsoo.



Kuva 16. Unityn animaatioasetuksia (Unity 2020)

3.1.4 Interaktiivisuus

Elokuvissa katsoja katsoo ennalta määriteltyä tarinaa ja animaatiota. Ohjaajalla ja animaattoreilla on täysi kontrolli siitä, millaisen kokemuksen katsoja saa. Videopelissä pelaaja kontrolloi hahmoa ja määrittelee, mitä tämä tekee ja miten tämä liikkuu. On olennaista, että hahmon ohjaus tuntuu pelaajasta miellyttävältä ja luonnolliselta.

Pelituotannossa on myös tärkeää välitön palaute pelaajalle. Jos pelaaja painaa hyppynappulaa, on hypyn tapahduttava välittämättömästi. Elokuvatuotannossa näitä rajoitteita ei ole, koska elokuvan katsominen ei ole interaktiivinen kokemus.

3.2 Pelin visuaalisen tyylin vaikutus animaatioon

Pelin visuaalinen tyyli, kuten myös sen genre, vaikuttavat myös animaatioon. Realistisessa kauhupelissä ei yleensä ole toivottua, että animaatiot ovat liioiteltuja ja karikatyyrisiä. Jos taas peli on tyyliltään erittäin minimalistinen ja yksinkertainen, liian levottomat animaatiot voivat olla häiritseviä.

Tarkastelen kolmea esimerkkipeliä, joiden visuaalinen tyyli on vaikuttanut animaation tyyliin.

3.2.1 Dead by Daylight

Dead by Daylight (kuva 17) on epäsymmetrinen kauhupeli, jossa neljän pelaajan selviytyjätiimi pelaa viidettä, tappajaa kontrolloivaa pelaajaa vastaan. Selviytyvät yrittävät korjata generaattoreita ja näiden avulla aktivoida portit, joiden kautta he pääsevät pakoon. Tappajan tavoite on estää selviytyjiä korjaamasta generaattoreita, jahdata selviytyjiä ja saadessaan nämä kiinni seivästää heidät koukkuun.

Kuva 17. Dead by Daylight (Behaviour Interactive Inc 2016)

Dead by Daylight on teemaltaan ja tunnelmaltaan pelottava ja ahdistava sekä tyyliltään realismiin kallistuva. Tämä heijastuu myös animaatioon. Liikkeet ovat hienovaraisia ja oikeaa elämää imitoivia. Koska peli on sävyltään vakava, animaatioita ei ole liioiteltu tai karikatoitu.

3.2.2 Untitled Goose Game

Untitled Goose Game on peli, jossa pelaaja ohjaa hanhea, jonka tavoitteena on kiusata ja härnätä paikallisen kylän asukkaita mahdollisimman paljon. Hanhella on tehtävälista erinäisistä kiusanteoista, joita pelaajan on määrä toteuttaa.

Peli on erittäin tyylitelty ja myös minimalistinen. Sekä hahmot että ympäristö koostuvat yksinkertaisista ja selkeistä muodoista. Hahmojen väritys on tasaista ja varjot lähes olemattomia. Ihmishahmot ovat myös kasvottomia.

Luonnollisesti myös pelin animaatiot ovat selkeitä ja virtaviivaisia. Koska peli on sävyltään koominen, ovat ihmishahmojen liikkeet liioiteltuja ja karikatyyrimäisiä,

jopa dramaattisia (Kuva 18). Tämä on tärkeää myös, jotta pelaaja pystyy selkeästi ymmärtämään ihmishahmojen reaktion hanhen kepposiin ja näin päättelemään, miten kannattaa toimia, jotta tehtävälistan ilkityöt saadaan hoidettua.

Kuva 18. Untitled Goose Game (House House 2019)

3.2.3 Minecraft

Minecraft on selviytymispeli, jossa kerätään erilaisia materiaaleja, joita pelaaja voi käyttää hyödyksi rakentamalla uusia materiaaleja ja tavaroita (kuva 19). Pelaajan on pidettävä huolta nälästään, ja energiastaan, sekä taisteltava monenlaisia eri vihollistyyppejä vastaan. Pelistä on myös versio, joka on omistettu nimenomaan rakentamiselle, ilman selviytymiselementtiä.

Kuva 19. Minecraft (Minecraft 2021)

Minecraftin tyyli on yksinkertaisuudessaan nykyään ikoninen. Hahmot ja ympäristöt muodostuvat kokonaisuudessaan erilaisista kuutioista. Minecraftin äärimmäinen yksinkertaisuus ilmenee myös animaatioissa: hahmoilla tai eläimillä ei ole lähes minkäänlaista persoonallisuutta luovaa animaatiota. Suurin osa hahmoista ainoastaan kävelee tai seisoo. Hahmot liikkuvat jäykästi heiluttaen suoria raajojaan.

4 Animaatiosyklien prosessi kettuhahmolla

Tämä projekti on tehty Frozenbyten videopeliprojektiin. Työtehtäväni oli tehdä kettuhahmolle animaatiosyklit ja varmistaa, että ne toimivat yrityksen pelimoottorissa. Ketun rigattu 3D-malli annettiin minulle valmiina, keskityin itse vain animaatioon.

4.1 Tutkimustyö ja referenssien etsiminen

Tässä projektissa on tavoitteena lajityypillinen, mutta kuitenkin persoonallinen käyttäytyminen. Jotta hahmon, tässä tapauksessa ketun, liikkeisiin saadaan tätä

luontevaa ja lajityypillistä käytöstä ja liikettä, on aluksi tärkeää etsiä referenssimateriaalia. Youtube on nykyään erinomainen työkalu referenssivideoiden etsimiseen.

Joskus sopivaa referenssiä ei löydy tai video on liian huonolaatuinen ollakseen kunnolla hyödyksi. Kettu on villieläin, joten siitä löytyvä materiaali ei ole välttämättä parhaan laatuista. Kettu ja koira ovat kuitenkin ruumiinrakenteeltaan lähellä toisiaan, joten käytin tässä projektissa myös videoita koirista referenssimateriaalina, sillä niistä on paljon erittäin hyvää ja selkeää kuvamateriaalia.

Koska tämän projektin tyyli on jotain realismin ja tyylitellyn fantasian välimaastolla, keskityn myös animaatiossa tällaiseen tyyliin. Animaatiot ovat hieman liioitellumpia kuin ehkä oikeassa elämässä, mutta ne kuitenkin pohjautuvat realismiin, ja niiden tarkoitus ei ole olla karikatyyrimäisen yliampuvia. Haluan lähinnä lisätä persoonaa ja inhimillisyyttä eläinhahmoon. Koska kyseessä on pelianimaatio, on olennaista, että animaatiot ja niiden tarkoitukset ovat selkeitä katsojalle ymmärtää.

4.2 Syklien animointi

Syklien tärkein ominaisuus on niiden toistuvuus. Animaatio päättyy samaan asentoon mistä se alkoi, jotta liike voi luontaisesti toistua peliympäristössä useaan kertaan.

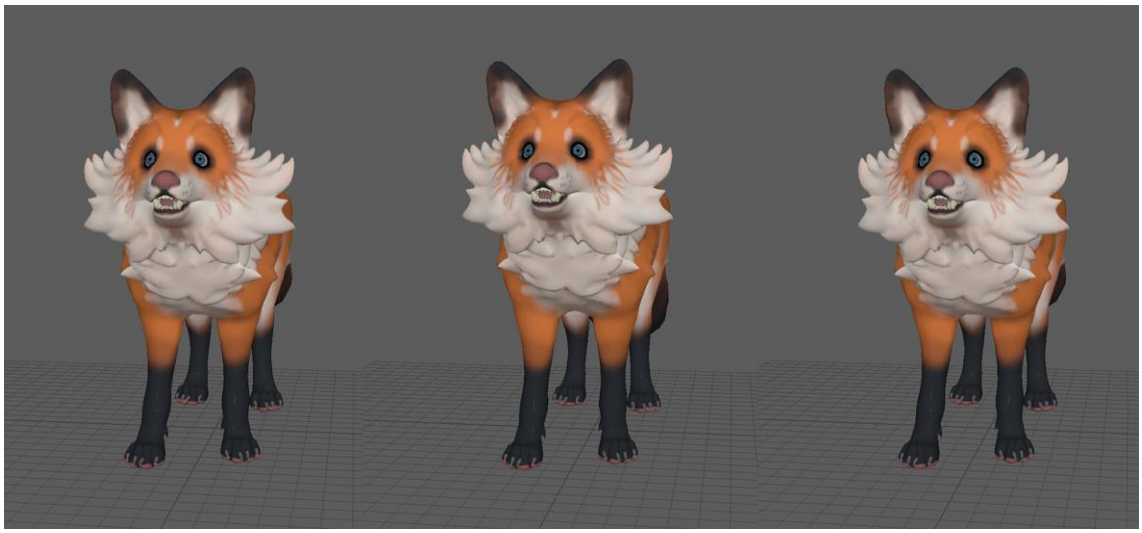
Toistuvasta liikkeestä tulee helposti mekaaninen ja liian ilmiselvästi itseään toistava. Yksi keino poistaa tätä mekaanisuutta, on tehdä animaatioista variaatioita, jotka ohjelmoidaan pelimoottorissa toistumaan sattumanvaraisesti. Näin hahmo ei toista yhtä ja samaa liikettä. Vaihtelemalla rauhallisempia ja aktiivisempia animaatioita, saadaan kokonaisuudelle miellyttävä tasapaino.

Tässä projektissa päätin tehdä kettuhahmolle yhden yksinkertaisen seisomisanimaation, joka toimii rauhallisena pohjana sen variaatioille. Variaatioiksi muodostuivat kettu katselemassa ympärilleen sekä kettu nuuskimassa maata. Nämä

animaatiot muodostavat yhdessä seisovan ketun kokonaisuuden. Tämän lisäksi animoin ketun kävelemään ja juoksemaan.

4.2.1 Seisominen

Tämä on ketun perusanimaatio, kun se seisoo paikallaan (kuva 20). Animaation on tarkoitus olla erittäin rauhallinen ja minimalistinen. Jos hahmo liikkuu koko ajan suurin elein, se käy herkästi katsojalle raskaaksi ja kokonaisuudesta tulee levoton tunnelma. Koska seisova kettu on rentoutunut ja rauhallinen, se on myös tunnelma, jota haluan ilmaista tämän animaatioissa.



Kuva 20. Seisova kettu

Animoin ketulle hengityksenomaista liikettä. Sen koko keho kääntyy hellästi edestakaisin, ja suu avautuu ja sulkeutuu hengitysten mukana. Tällä animaatiolla ei ole tarkoitus olla varsinaista persoonaa, se on nimenomaan pohja variaatioille, jotka seuraavaksi teen.

4.2.2 Ympärille katselu

Tämä animaatio on variaatio ketun seisomisanimaatioon. Halusin variaation, joka ei muuta dramaattisesti ketun seisovaa asentoa, vaan hienovaraisesti rikkoo yksinkertaista seisovaa hengitystä. Tässä animaatiossa kettu katsoo rauhallisesti

ympärilleen, kuulee äänen, ja kääntyy katsomaan äänen suuntaan. Lopuksi kettu palaa perusasentoon. (Kuva 21)

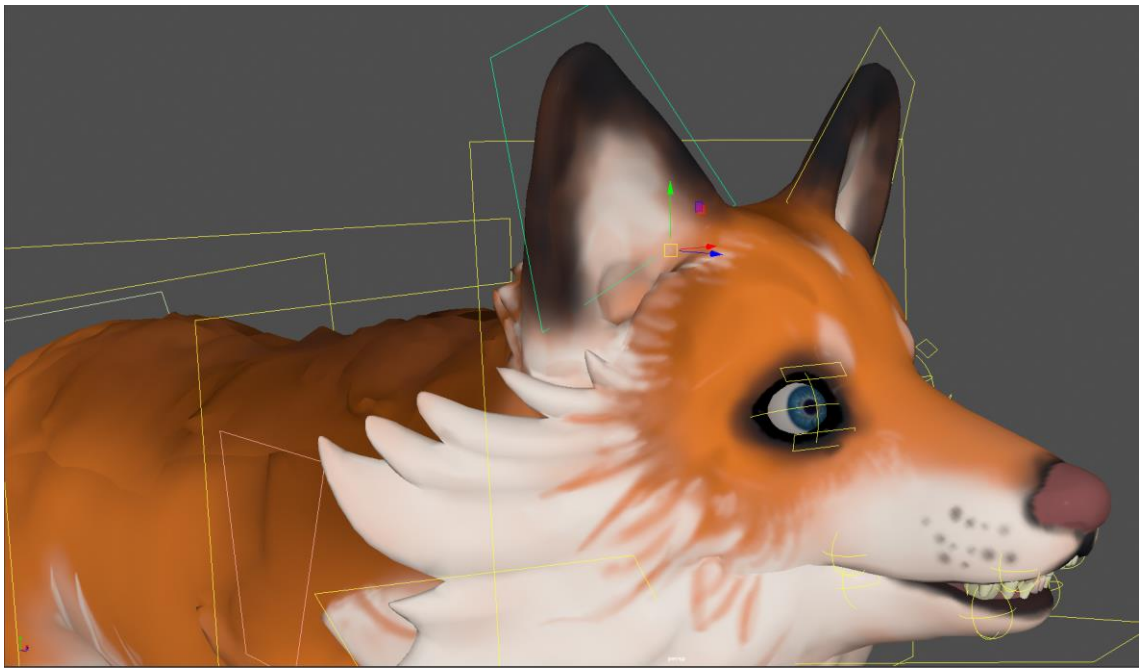
Kuva 21. Ympärilleen katseleva kettu

Animaatiossa nopeus ja sen vaihtelu on tärkeää. Jos animaatio on tasapaksua, etenkin jos se on tasaisen hidasta, se tuntuu tylsältä ja elottomalta. Tämän takia keskityin liikkeen nopeuden variaatioihin. Ensin kettu kääntää päänsä nopeasti vasemmalle, liike rauhoittuu, kun tämä katselee ympärilleen, sitten kettu kuulee jotain oikealla, ja kääntää vikkelästi pään siihen suuntaan. Tämän jälkeen animaation annetaan taas rauhoittua, kun kettu katsoo ja kuulostelee äänen kohdetta, ja lopulta se palautuu takaisin alun asentoon.

Itselleni on haastavaa muistaa antaa rauhallisille hetkille tarpeeksi aikaa. Koska tässä animaatiossa rauhallisuus oli olennainen osa kokonaisuutta, oli tarpeellista kiinnittää erityistä huomiota, etten kiirehtinyt asennosta toiseen liian nopeasti.

Tärkeää on myös varmistaa, ettei rauhallinen animaatio, hidas hengittely, muutu yhtäkkiä nopeaksi liikkeeksi. Tämä tekisi animaatioiden blendauksesta hankalaa ja epäluonnollista, ja kiinnittäisi myös katsojan huomion epämiellyttävällä tavalla. Tämän takia animoin ketun ensimmäisistä liikkeistä ja asennon muutoksista melko pieniä. Myöhemmin animaatiossa liikettä ja muutoksia tapahtuu enemmän, koska uudelle liikkeelle on annettu aikaa käynnistyä.

Ketun korvassa oli vain yksi luu (kuva 22), joten itse korvaa ei pysty taivuttamaan mikä on sääli, koska liikkeen mukana taipuvat korvat näyttävät hyvältä. Tämä luiden puute teki myös korvan liikkeestä, kun kettu kuulee jotain, hankalan toteuttaa. Vaikka liike tuntuu ja näyttää yksinkertaisella, on luontevan liikkeen tekeminen, kun vain yksi osa kehosta liikkuu, haasteellista. Lopulta onnistuin kuitenkin saamaan liikkeen mieleisekseni.



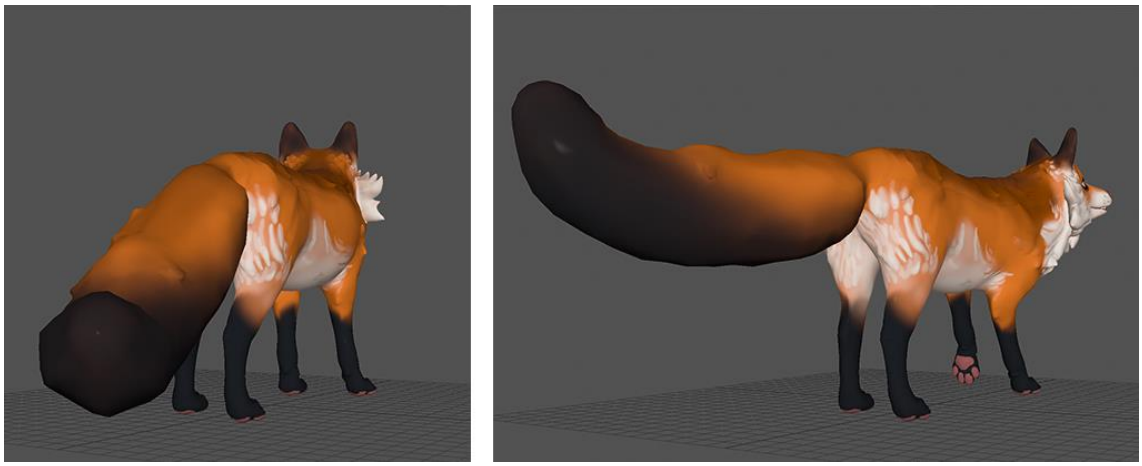
Kuva 22. Ketun korvakontrolli

Kaikissa animaatioissa, jossa teemana on hahmon katse, ovat silmät luonnollisesti olennainen osa. Oikeassa elämässä, kun katsomme ympärillemme, eivät silmät pysy paikallaan tai liiku täysin pään liikkeiden mukana. Tämän vuoksi animoin ketun silmät vilkuilemaan ensin alas ja sitten ylös, kun se koittaa pohtia, mistä tämän äsken kuulemma ääni tuli.

Animaatiossa hahmot eivät räpäytä silmiään turhaan. Silmien räpäytyksellä on aina joko emotionaalinen tai liikkeen kannalta olennainen funktio. Tämä on totta myös tässä animaatiossa, jossa animoin ketun räpäyttämään silmiä nimenomaan kohdissa, jossa kettu kääntää päätään (kuva 23). Tämä siksi, että yksi näistä räpäytyksen mahdollisista tarpeellisista funktioista on nopeat katseen suunnan vaihdokset.

Kuva 23. Kettu räpäyttää silmiään kääntyessään

Eläimillä yksi suuri tunteiden ilmaisukeino joita ihmishahmoilla ei ole, on häntä. Eläimet ilmaisevat tunteitaan hännän kautta monella eri tavalla. Tämän takia, kun kettu yllättyy äänestä, jonka se kuulee, animoin ketun hännän nousemaan pystyasentoon (kuva 24). Kettu kuulostelee valppaana, mistä on kyse. Kun kettu rauhoittuu takaisin perusasentoon, niin laskeutuu häntäkin.



Kuva 24. Häntää nostava kettu

Animoin myös hännän kääntymään aina päätä vastakkaiseen suuntaan. Kun kettu kääntyy vasemmalle, häntä kääntyy oikealle. Tämä luo miellyttävää vaihtelua ketun liikkeisiin, sen sijaan että jokainen liike kohdistuisi samaan suuntaan.

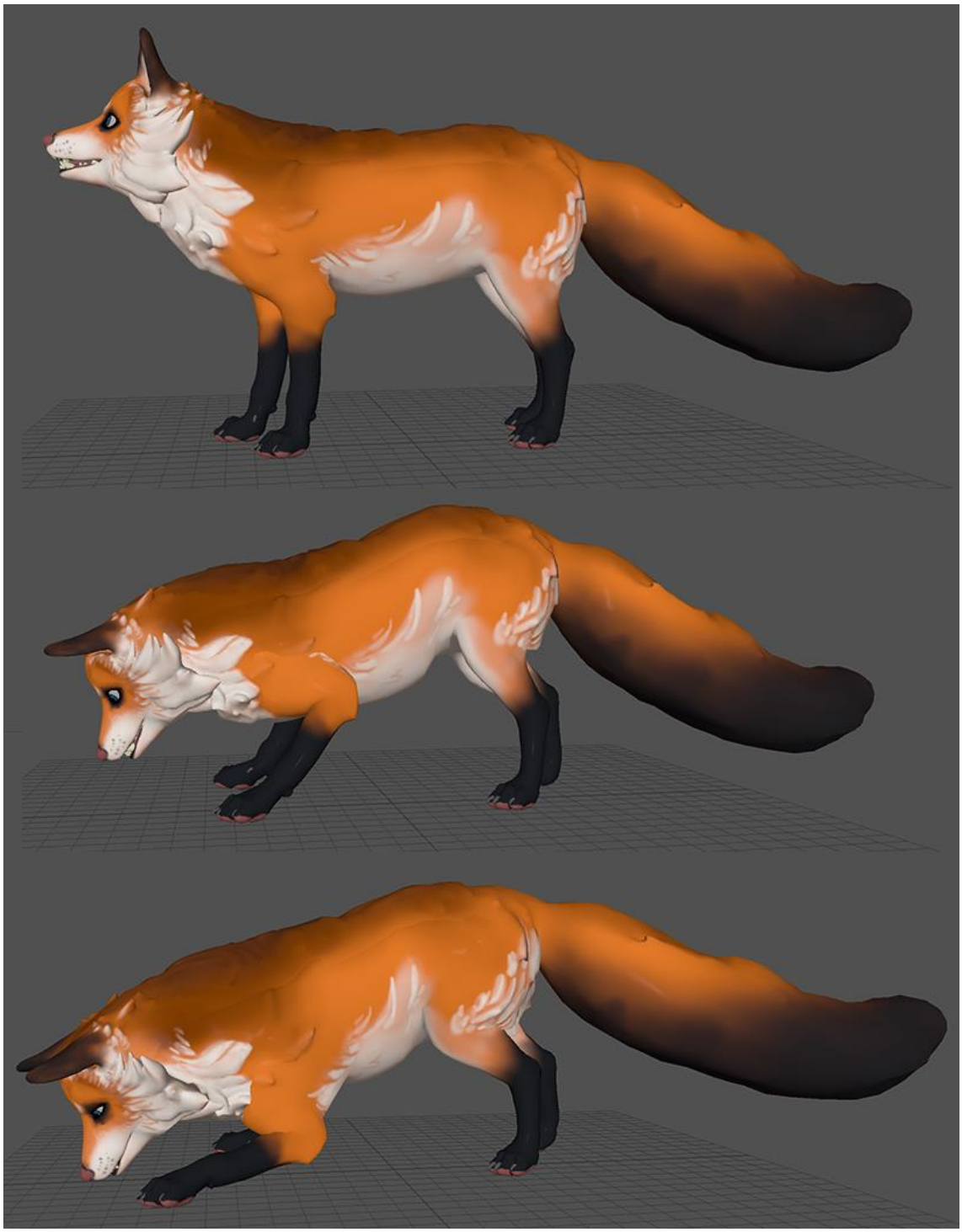
4.2.3 Nuuskiminen

Nuuskiminen on toinen variaatio ketun seisomisanimaatioon. Tällä kertaa ketun perusasento rikkoutuu enemmän, ja asennon muutos on ympärille katsomisanimaatiota dramaattisempi. Kettu katsoo alas, ja kumartuu sitten nuuskimaan maata. Lopuksi se nousee takaisin perusasentoon (kuva 25).

Kuva 25. Nuuskiva kettu

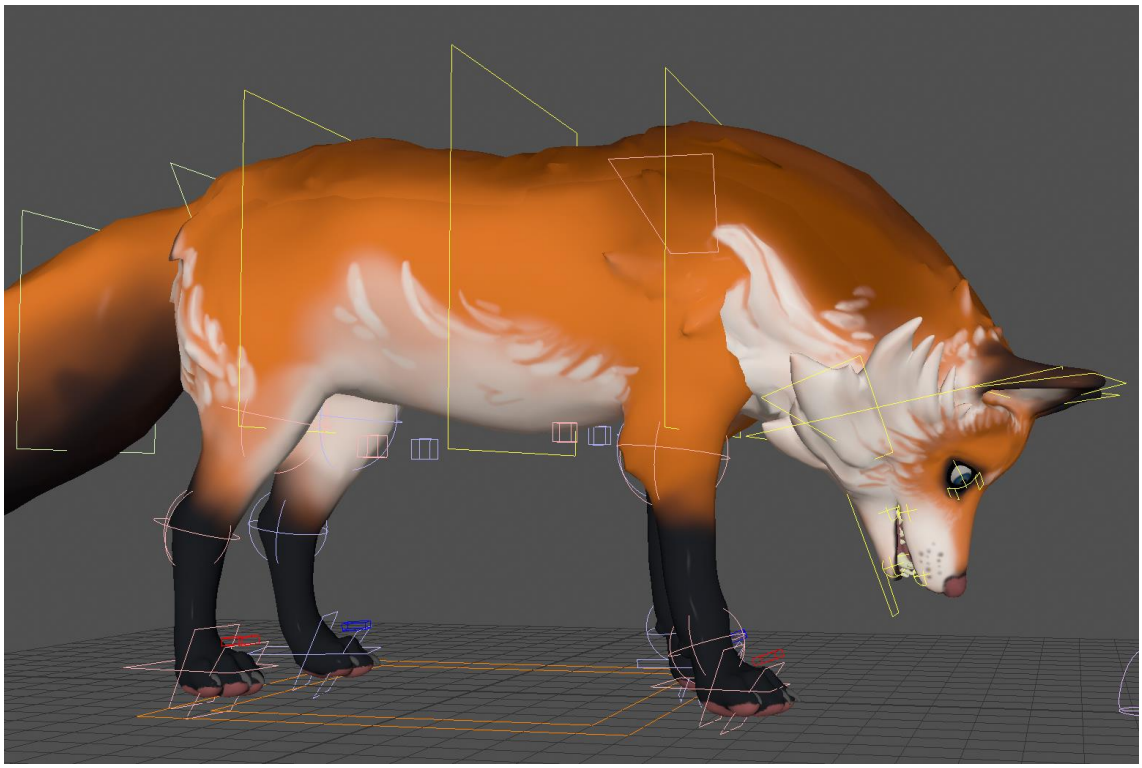
Animaatioissa, joissa hahmo vaihtaa merkittävästi asentoa, on haasteena luonnollinen palaaminen perusasentoon. Sille on annettava tarpeeksi aikaa, tai se on töksähtävä ja luonnoton. Olennaista on, että hahmo ei vaihda asentoa perusasentoon yhtäaikaisesti, vaan eri raajat liikkuvat eri tahtiin. Animoin itse ketun hännän siirtymään loppuasentoon ensin, jota seurasivat keho ja jalat. Lopuksi hahmon pää asettuu paikoilleen.

Kun kettu kumartuu, sen painopiste siirtyy eteenpäin. Ketun etujalat vastaanottavat suurimman osan painosta. Kehon siirtäminen painopiirteen mukaisesti on itselleni haastavaa, ja jouduin kokeilemaan useamman kerran, kunnes sain sen mieleisekseni (kuva 26).



Kuva 26. Kumartuvan ketun painopiste

Kettu kumartuu liioitellummin kuin kettu oikeassa elämässä ehkä tekisi. Animoin ketun kumartumaan eteenpäin koko ruumiillaan, sillä pelkän kaulan taivuttaminen ei näyttänyt hyvältä, vaan aiheutti epäluonnollisen näköisen paisuneen mutkan hahmon kaulaan (kuva 27).



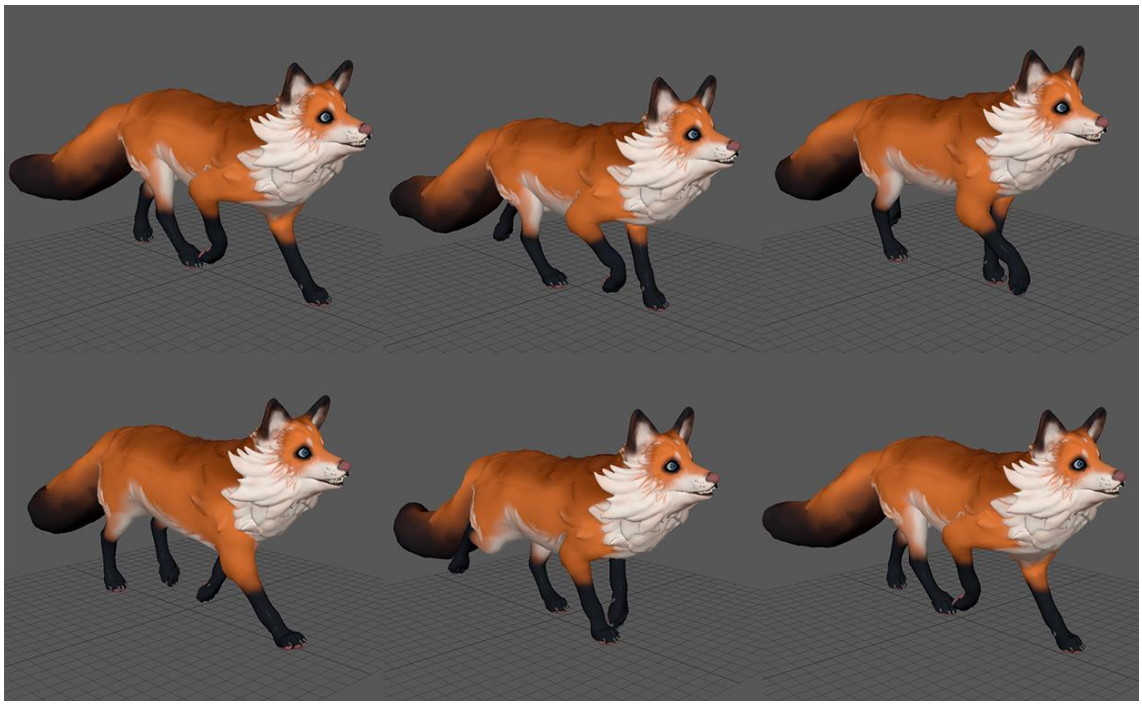
Kuva 27. Ketun kaula taipuu epäluonnollisesti

Jotta animaatio oli mielenkiintoisempi, lisäsin siihen osan, jossa nuuskiva kettu astuu eteenpäin ja korjaa asentoaan päästäkseen paremmin nuuskimaan.

Animoin jälleen tarkoituksen omaavaa silmien räpyttelyä, ensiksi kun kettu kumartuu maahan, ja toisen kerran, kun kettu vaihtaa asuntoa nuuhkiessaan. Kettu räpäyttää myös silmiä palatessaan loppuasentoon.

4.2.4 Kävely

Kävely (kuva 28) on yksi niistä animaatioista, joihin käytin apuna referenssimateriaalia koirista kettujen sijaan. Kävelyanimaatiot ovat pelianimaatiossa standardi, mutta nelijalkainen hahmo tekee luonnollisesti animaatiosta haastavampaa.



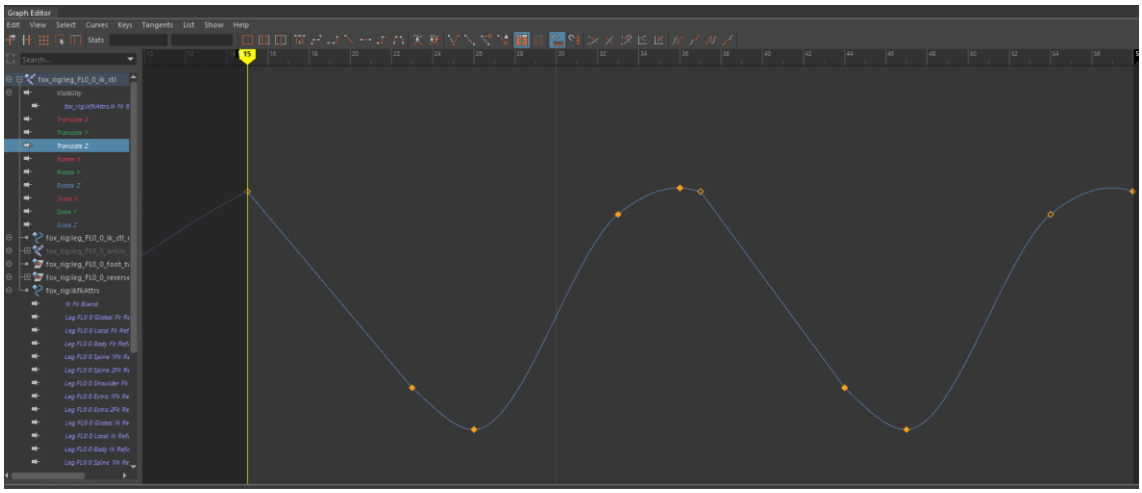
Kuva 28. Kävelevä kettu

Aloitan kävelyanimaatiot aina animoimalla jalat ensin. Sen jälkeen lisään kehon liikkeet, kehon rotaatiot, ja lopuksi yksityiskohdat.

Tein animaatiosta aluksi liian hitaan. Kettu käveli todella verkkaisesti, ja kun sen laittoi kulkemaan eteenpäin, huomasin että se kävelee eteenpäin aivan liian hitaaseen tahtiin. Nopeutin animaatiota ja tein kävelystä vähän ripeämmän.

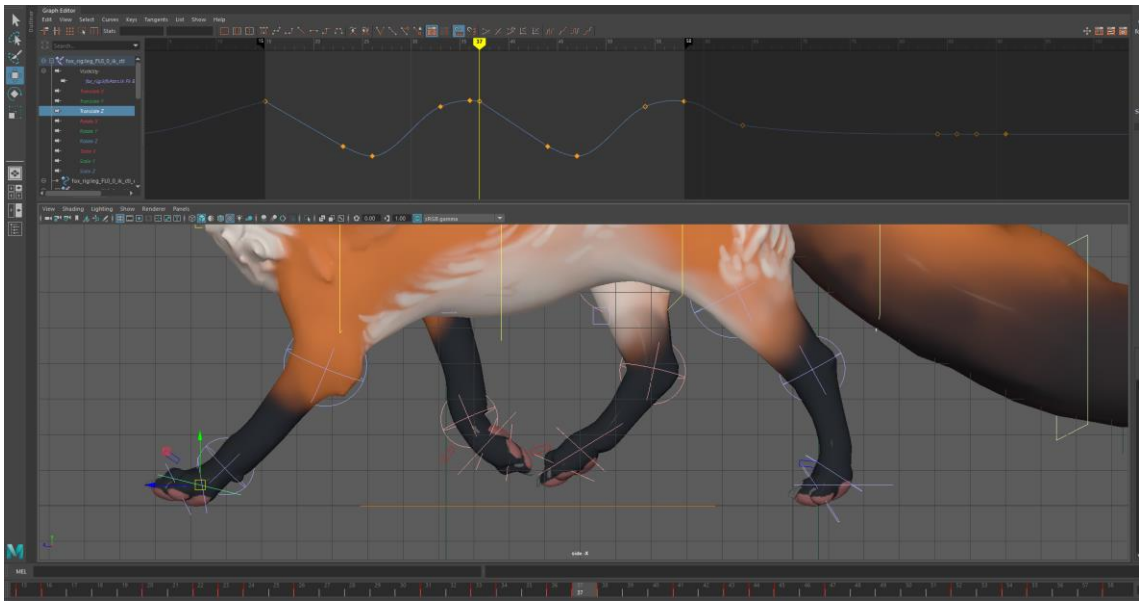
Animoin myös aluksi ketun kävelemään jalat liian levällään. Ketut, ja useat muut eläimet, kävelevät jalat todella lähellä keskiakselia.

Kun tassu osuu maahan ja alkaa siirtyä taaksepäin hahmon liikkuessa eteenpäin, oli tärkeää, että tämä liike on täysin lineaarinen. Käytin animaatioiden tekemiseen Maya-ohjelmaa, ja se luo automaattisesti animaatiot tavalla, jossa ne alkavat ja loppuvat hitaammin, ja kiihtyvät liikkeen keskellä (kuva 29). Tämä on yleensä toimiva systeemi, mutta jos tassut liikkuvat maassa ollessaan joissain kohdin hitaammin kuin toisessa, hahmo alkaa liukua, sen sijaan että sen jalat olisivat tukevasti maassa.



Kuva 29. Lineaarin ja automaattisen liikekaaren ero

Hahmon liikenopeus saadaan laskemalla yhteen tassujen position muutos. Ketun tassu aloittaa astumaan maahan positiossa 40 ja nousee maasta positiossa -23. Näin ollen kettu kävelee nopeudella 70 (kuva 30).



Kuva 30. Ketun tassujen positio

Kävely jäi herkästi jäykäksi ja luonnottomaksi. Poistaakseni tätä mekaanisuutta, keskityin siihen, että ketun ruumis on käytännössä kahdessa eri osassa. Animoin ketun etuosan liikkumaan eri tahtiin kuin takaosan. Tällainen epäsymmetrinen liike on hyvä tapa tehdä liikkeestä luonnollisempi.

4.2.5 Juoksu

Juoksuanimaatio (kuva 31) oli toinen animaatio, johon katsoin referenssiä koiravideoista. Tästä tulikin myöhemmin itse asiassa ongelma, liioittelin ketun selän taipumista aivan liikaa. Kettujen selkä on juostessa ja kävellessä lähes suora.

Kuva 31. Juokseva kettu

Prosessi on käytännössä hyvin samanlainen kuin kävelyanimaatiota tehdessä. Itselleni on todella haastavaa saada juoksuanimaatiosta luonnollinen ja sulava. Juoksuanimaatiossa on frameja niin vähän, että tuntuu hankalalta löytää tarpeeksi aikaa lisätä ylimääräistä animaatiota esimerkiksi korviin tai häntään.

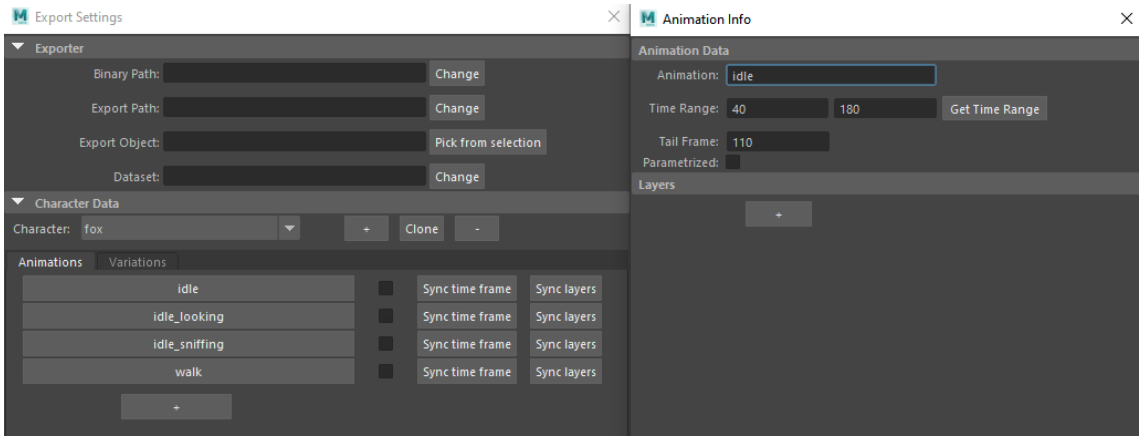
Animaation pituus riippuu sen nopeudesta. Ketun juoksuanimaatio on 15 freimiä mutta myös esimerkiksi 7 freimin juoksuanimaatio on mahdollista tehdä. Tekemäni kävelyanimaatio taas oli 22 freimiä.

4.3 Animaatioiden lisääminen pelimoottoriin

Animaatioiden lisäämisen prosessi riippuu siitä, mitä ohjelmaa käytetään. Omassa tapauksessani käytin työnantajani Frozenbyten omaa pelimoottoria. Tämän pelimoottorin prosessiin kuuluu FBX-tiedostojen teettäminen animaatioista, LUA-skriptin kirjoittaminen ja lopuksi pelimoottorissa testaaminen.

4.3.1 Tiedostojen valmistelu

Koska animoin kaikki syklit samaan tiedostoon, määrittelen ensin Mayassa, ohjelmassa, jolla animoin, mistä freimistä käytettävä animaatio alkaa ja loppuu. Käytän Frozenbyten omaa pelimoottoria, ja sitä varten tulee määritellä animaatiolle myös tail framen, eli kohdan, jossa animaatio alkaa blendautumaan toiseen (kuva 32). Tämä varmistaa, että animaatiot vaihtuvat sulavasti keskenään. Tail frame määritellään yleensä kohtaan, jossa liike on juuri rauhoittumassa takaisin loppuasentoon. Koska juoksu- ja kävelyanimaatiossa ei tällaista rauhoittumista ole, kopioin niiden kohdalla liikkeen toistumaan kahdesti peräkkäin ja määrittelin tail frameksi ensimmäisen freimin, jossa liike on alkamassa toistumaan. Näin liikkeillä on tarpeeksi aikaa blendautua. Jos käyttäisin esimerkiksi Unity-pelimoottoria, tekisin tämän osuuden Blend Tree -ominaisuutta käyttäen.



Kuva 32. FBX-tiedostojen asetukset

Animaatiot ovat siirrettävä muotoon, jossa pelimoottori tunnistaa ne. Tämän takia teetän animaatioista FBX-tiedostot. Tämä formaatti tallentaa olennaiset tiedot kuten 3D-mallin ja animaation ja toimii oikein käyttämässäni pelimoottorissa.

4.3.2 Animaatiokontekstien luominen

Koodaan LUA-skriptikielellä animaatiolle omat kontekstit (kuva 33), jossa määrittelen ominaisuudet, joita pelimoottori tarvitsee. Käytän esimerkkinä nuuskimisanimaatiota, skriptissä käytän englanninkielistä termiä sniffing. Ensin luon animaatiolle addState-koodin avulla tilan eli määrittelen, minkälainen animaatio on kyseessä. Tässä tapauksessa kyseessä on idle-animaatio, eli käytännössä hahmo seisomassa, tekemättä mitään. Seuraavaksi BlendTime-koodilla määrittelen kuinka nopeasti animaatio blendautuu muihin animaatioihin, käyttäen hyväksi tail frame-informaatiota.

```
--{ Animation: s_idle_sniffing
s_idle_sniffing = addState(idle, "s_idle_sniffing");
s_idle_sniffing:blendTime(200);
s_idle_sniffing:loop(false); -- true
s_idle_sniffing:restart(true);
s_idle_sniffing:selectionWeight(2);
s_idle_sniffing:speedFactor(1.0);
s_idle_sniffing:add("idle_sniffing.fbxanim");
--}
```

Kuva 33. Nuuskimisanimaation konteksti

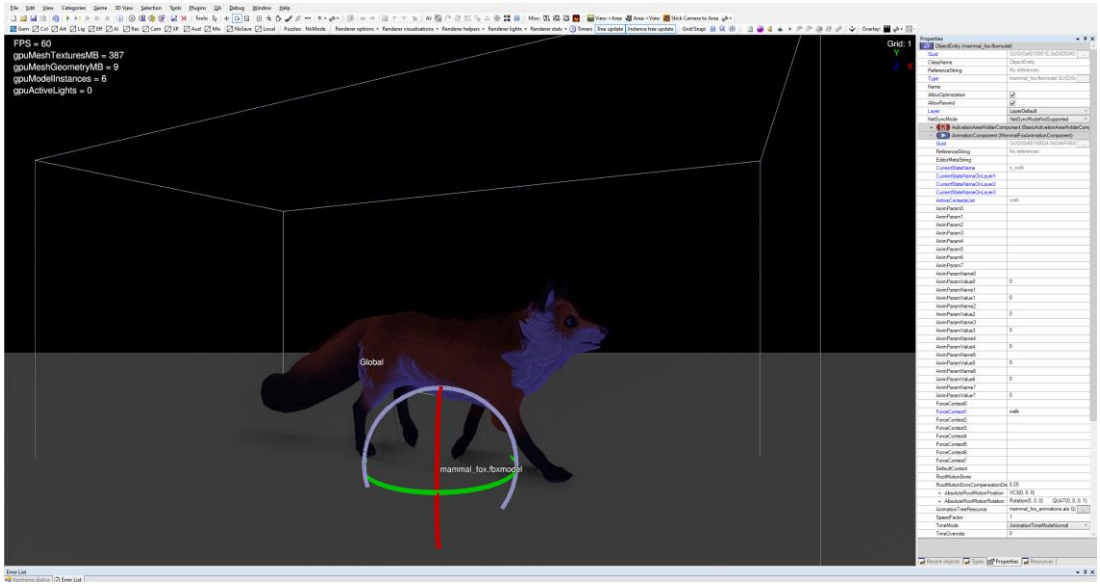
Määrittelen myös loop-koodilla onko, animaatio toistuva animaatio eli toistuuko se heti uudestaan animaation loputtua. Vaikka sykli itsessään on tehty toistumista varten, nuuskimisanimaatio on vain yksi osa kokonaisuutta. Seisoen hengittely, nuuskiminen ja ympärilleen katsominen toistuvat kaikki satunnaisesti keskenään. Tämän takia merkitsen, että yksittäinen nuuskimisanimaatio ei toista itseään. Restart-koodi kertoo, saako kyseinen tila alkaa uudestaan, kun se on juuri toistunut. Eli jos hahmo nuuskii, sallinko satunnaisen valinnan osua nuuskimiseen uudestaan.

SelectionWeight määrittelee, kuinka suurella todennäköisyydellä tämä animaatio tulee valituiksi muista idle-tilan animaatioista. Mitä isompi luku, sitä suuremmalla todennäköisyydellä kyseinen animaatio toistuu. Esimerkiksi nuuskimisanimaation arvo on 2, ja hengittelyn arvo on 13. Koska nuuskiminen on variaatio hengittelylle, sitä on tarkoitus nähdä paljon harvemmin. SpeedFactorin avulla animaatioita voi nopeuttaa tarpeen mukaan. Nuuskimisanimaatiota ei tarvitse nopeuttaa, joten pidän sen arvona 1.0. Juoksuanimaation arvo on 1.1, sillä se kaipasi vähän lisää vauhtia.

Lopuksi vielä määrittelen mitä FBX-animaatiotiedostoa kyseinen animaatiokonteksti käyttää. Nuuskimisanimaation konteksti käyttää tietenkin tiedostoa, joka sisältää kyseisen animaation.

4.3.3 Pelimoottorissa testaaminen

Pelimoottorissa, jota käytän, animaatioita testataan lisäämällä niille manuaalisesti haluttu konteksti, esimerkiksi kävely (kuva 34). Kun tämän kontekstin vaihtaa, esimerkiksi nuuskimisesta ympärilleen katseluun, nämä kaksi animaatiota blendautuvat toisiinsa. Näin pystyn myös testaamaan, olenko asettanut tail framet sopivaan kohtaan animaatiossa ja näyttääkö blendautuminen luonnolliselta.



Kuva 34. Kävelevä kettu pelimoottorissa

Jos animaatio näyttää 3D-ohjelmassa hyvältä, lähes aina se näyttää myös tässä pelieditorissa hyvältä. Yleensä muutoksia kaipasivat eniten animaatioiden nopeudet, sekä kahden animaation välinen blendautuminen. Joskus tail framen sijainnin muokkaus riitti, joskus oli tarpeellista editoida itse animaatiota ja varmistaa, että lopussa animaatiolla on tarpeeksi aikaa rauhoittua.

Yleisin ongelma itselläni animaatioita testatessa oli, että olin kirjoittanut jotain väärin LUA-skriptiä tehdessäni. Ongelmana oli esimerkiksi, että osa animaatioista toimi, mutta osa ei. Välillä animaatio, jonka oli tarkoitus jäädä toistumaan ikuisesti, ei toistunutkaan. Kaikki animaatiot kuitenkin toimivat lopulta pelimoottorissa, eikä mitään suuria muutoksia tarvittu.

5 Pohdinta

Tavoitteeni oli kertoa asiat mahdollisimman yksinkertaisella ja ymmärrettävällä tavalla, jotta opinnäytetyö olisi helppo lukea ilman kokemusta 3D-animaatiosta tai pelituotannosta. Tämän takia käsittelin aihealuetta melko tiiviisti.

3D-animaation historiasta oli välillä hankala löytää informaatiota, sillä useimmat lähteet keskittyivät animaation historiaan kokonaisuudessaan, jolloin aiheena on enemmän traditionaalinen 2D-animaatio ja muut animaatiotekniikat. Halusin itse keskittyä nimenomaan 3D-animaatioon. Sain lopulta kuitenkin mielestäni kasattua selkeän kuvan siitä, miten 3D-animaatio kehittyi vuosien saatossa.

Välillä oli haastavaa löytää keino selittää monimutkaiset konseptit, esimerkiksi topologia tai rigaaminen, muutaman virkkeen avulla. Havainnollistavat kuvat tukevat kuitenkin tekstiä, ja auttavat lukijan ymmärtämistä.

Projektiosuudessa itselleni suurin haaste oli painopisteen huomioon ottaminen, kun hahmo liikkuu ja vaihtaa asentoa, sekä että osaisin antaa animaation rauhallisille hetkille tarpeeksi tilaa ja aikaa. Koen, että opin projektista paljon uutta ja nautin sen toteuttamisesta.

Lähteet

Animation Academy 2011. History of 3D Animation. http://multimedi-amcc.com/old-students/ashaver/3d_history.html (05.05.2021)

Autio, Ville & Hautala, Marika 2020. Pelialan haasteet ja monet mahdollisuudet. https://centriabulletin.fi/pelialan-haasteet-ja-monet-mahdollisuudet/ (17.05.2021)

Beegel, Justin 2020. What is 3D Animation? https://infographicworld.com/what-is-3d-animation/ (08.05.2021)

Designblendz Team 2019. What is 3D animation and how is it different from 3D animation? Designblendz Blog. https://www.designblendz.com/blog/what-is-3d-animation-and-how-is-it-different-from-2d-animation (05.05.2021)

Fernandes Da Conceição, Ricky 2019. The History of Super Mario: The N64 Days. Goomba Stomp. https://goombastomp.com/the-history-of-super-mario-the-n64-days/ (08.05.2021)

Kristel, 2017. Game Design Tutorial: How to Use Blend Trees in Unity. Eguca-tion & Technology Blog.https://www.studica.com/blog/game-design-tutorial-blend-trees-unity (18.05.2021)

Norman, Jeremy 2004. William Fetter First Uses a Computer to Create Three-Dimensional Images of the Human Body. https://www.historyofinfor-mation.com/detail.php?entryid=1041 (05.05.2021)

Pixelloid Studios 2018. What is the difference between 2d and 3D-animation? Pixelloid Blog. https://pixelloid.com/blog/what-is-the-difference-between-2d-and-3d-animation/ (05.05.2021)

Plantec, Peter 2012. The Fascinating Story of How Autodesk Came to Be (Part 1). https://www.studiodaily.com/2012/01/the-fascinating-story-of-how-autodesk-came-to-be-part-1/ (08.05.2021)

Pluralsight 2020. Animation for games vs animation for movies. https://www.pluralsight.com/blog/film-games/how-animation-for-games-is-different-from-animation-for-movies (08.05.2021)

PointInTimeStudios 2020. What is 3D Animation? https://pointintimestudios.com/what-is-3d-animation/ (05.05.2021)

Unity 2020. Blend Trees. Unity User Manual 2020.3 (LTS). https://docs.unity3d.com/Manual/AnimationOverview.html (15.03.2021)

5.1 Kuvalähteet

Allen, Peter 2017. Pirates, penguins, Wallace and Gromit … the Aardman show delights https://theconversation.com/pirates-penguins-wallace-and-gromit-the-aardman-show-delights-80195 (05.05.2021)

Andrias, Frenky 2018. 3D Layout animation showreel. https://www.youtube.com/watch?v=xbUqONGKs8k (05.05.2021)

Behaviour Interactive Inc 2016. Steam. https://store.steampowered.com/app/381210/Dead_by_Daylight/ (05.05.2021)

Durham, Colleen 2021. Cycle Library. https://colleenedurham.com/cycle-library (05.05.2021)

House House 2019. Untitled Goose Game Presskit. https://goose.game/presskit/ (05.05.2021)

Infinite Fall 2017. Night in the Woods. Steam. https://store.steampowered.com/app/481510/Night_in_the_Woods/ (05.05.2021)

introbrand 2018. From Toy Story to Terminator: The 3D Animation History. https://www.introbrand.com/blog/3d-animation-history/ (05.05.2021)

Kellet, Robert 2016. Remembering Crash Bandicoot (1996) – The Original Playstation Mascot. https://press-start.com.au/features/2016/05/10/remembering-crash-bandicoot-1996-original-playstation-mascot/ (05.05.2021)

Learoyd, Joseph 2019. Where Do Cartoons Come From? | The Process Of 3D Animation. https://www.headstuff.org/culture/visual-arts/where-do-cartoons-come-from-the-process-of-3d-animation/ (05.05.2021)

Minecraft 2021. About Minecraft. https://www.minecraft.net/en-us/about-minecraft (05.05.2021)

Moreno, Jean 2021. Toony Colors Pro 2. http://jeanmoreno.com/unity/toonycolorspro/ (05.05.2021)

Pile, Jonathan 2019. Kubo And the Two Strings Review. https://www.empireonline.com/movies/reviews/kubo-two-strings-review/ (05.05.2021)

Pramath 2017. Monster Hunter World Continues To Look Great In New 1080p Screenshots. https://gamingbolt.com/monster-hunter-world-continues-to-look-great-in-new-1080p-screenshots (05.05.2021)

Smithers, Dominic, 2020. Super Mario 64 Is Being Remastered. https://www.ladbible.com/news/gaming-super-mario-64-is-being-remastered-20200903 (05.05.2021)

Team Cherry 2017. Hollow Knight. Steam. https://store.steampowered.com/app/367520/Hollow_Knight/ (05.05.2021)

Unity 2020. Blend Trees. Unity User Manual 2020.3 (LTS). https://docs.unity3d.com/Manual/class-BlendTree.html (05.05.2021)

Liitteet

Ketun valmiit animaatiosyklit

kettu_animaatiosyklit.mp4